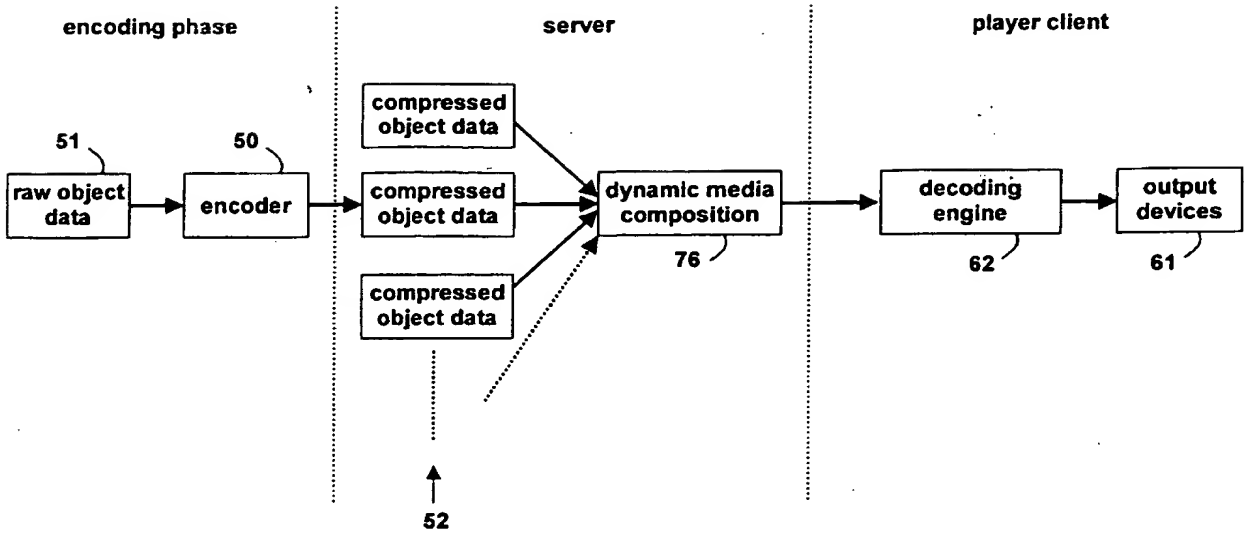

Figure 1

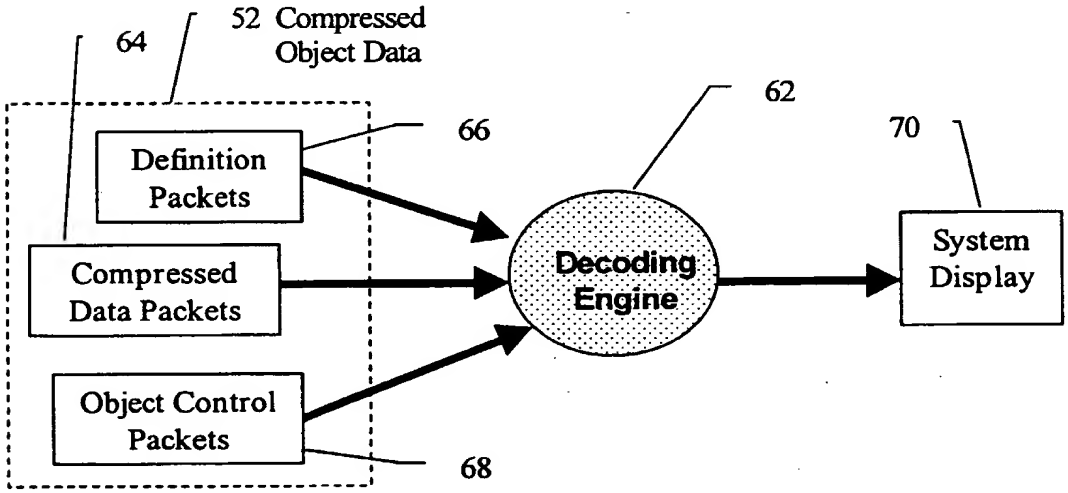

Figure 2

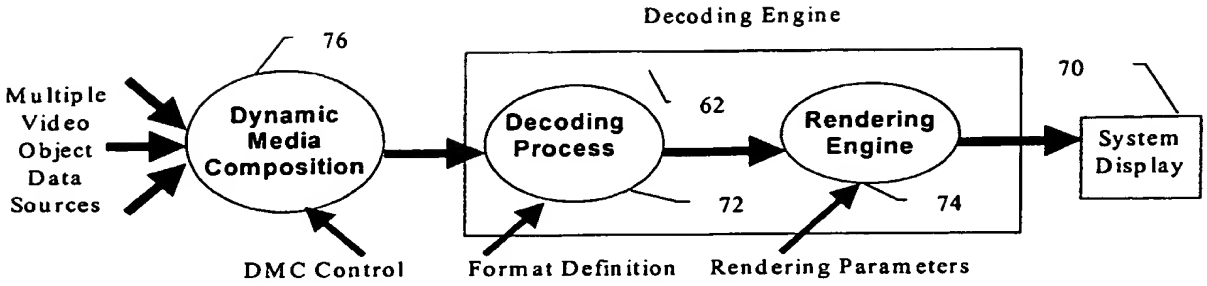

Figure 3

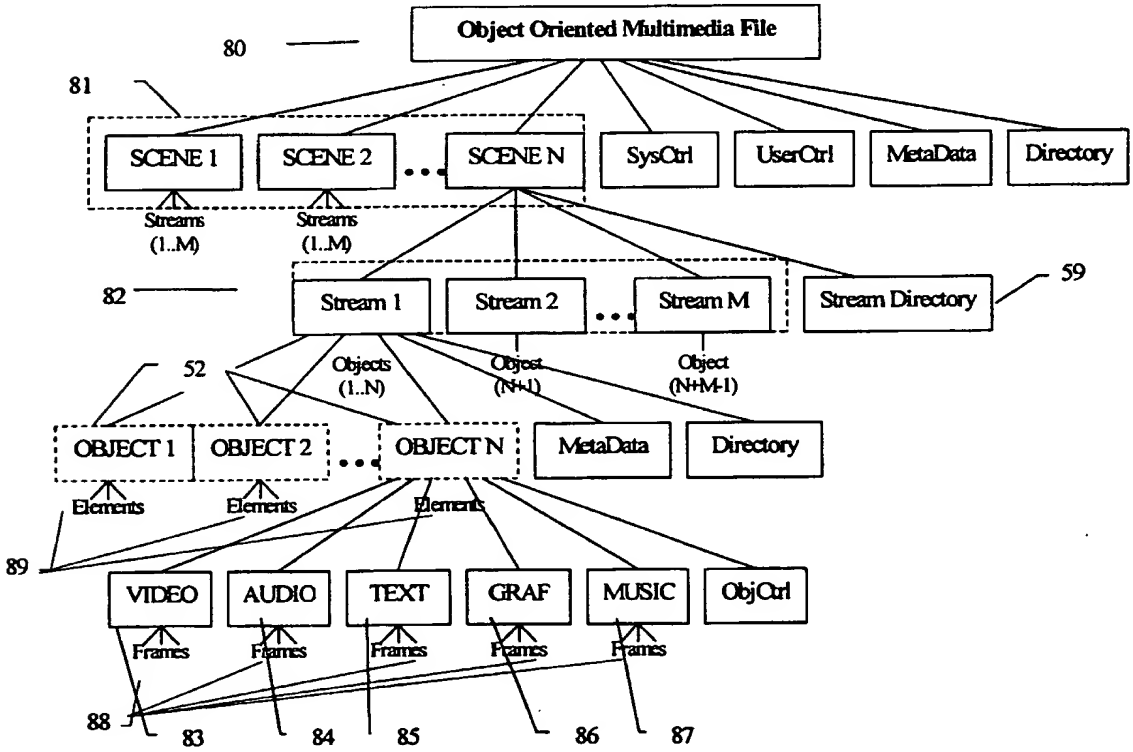

Figure 4

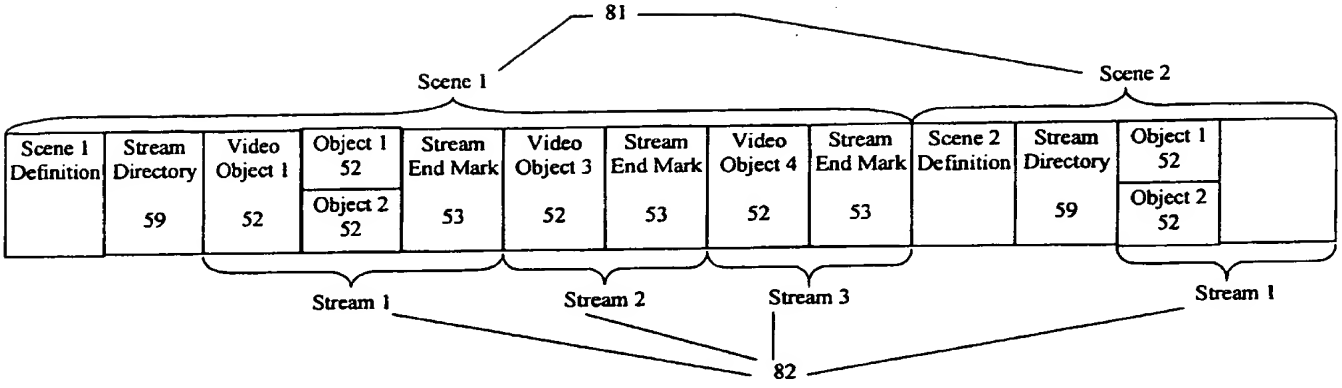

Figure 5

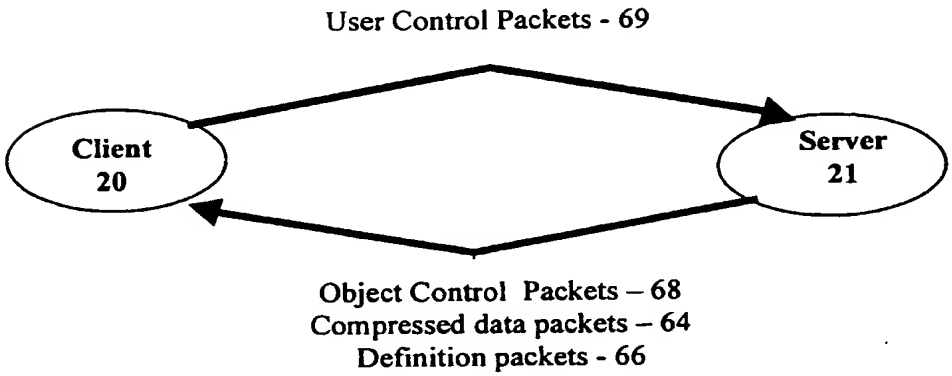

Figure 6

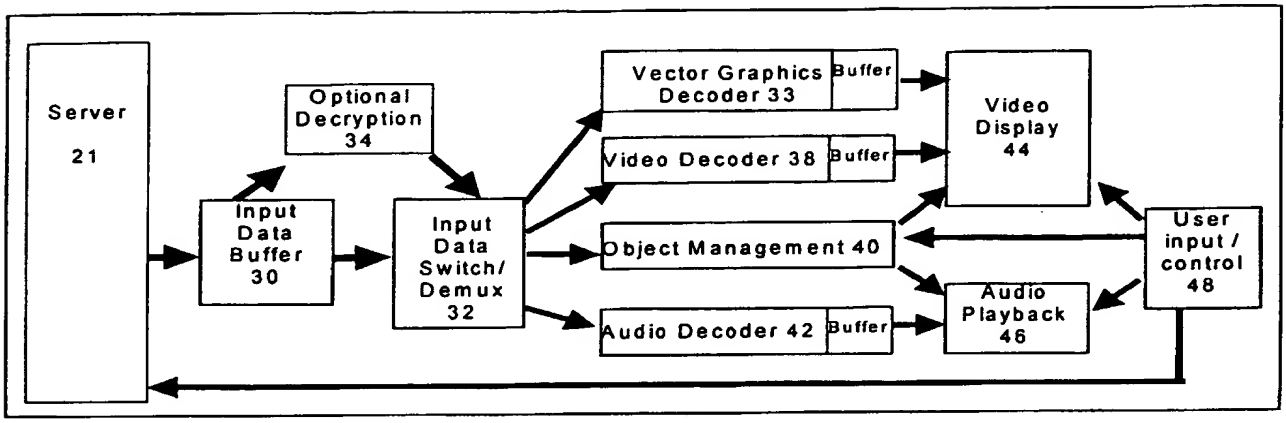
Server
21
Optional
Decryption
34
Input
Data
Buffer
30
Input
Data
Switch/
Demux
32
Vector Graphics
Decoder 33
Buffer
Video Decoder 38
Buffer
Object Management 40
Audio Decoder 42
Buffer
Video
Display
44
Audio
Playback
46
User
input /
control
48

Figure 7

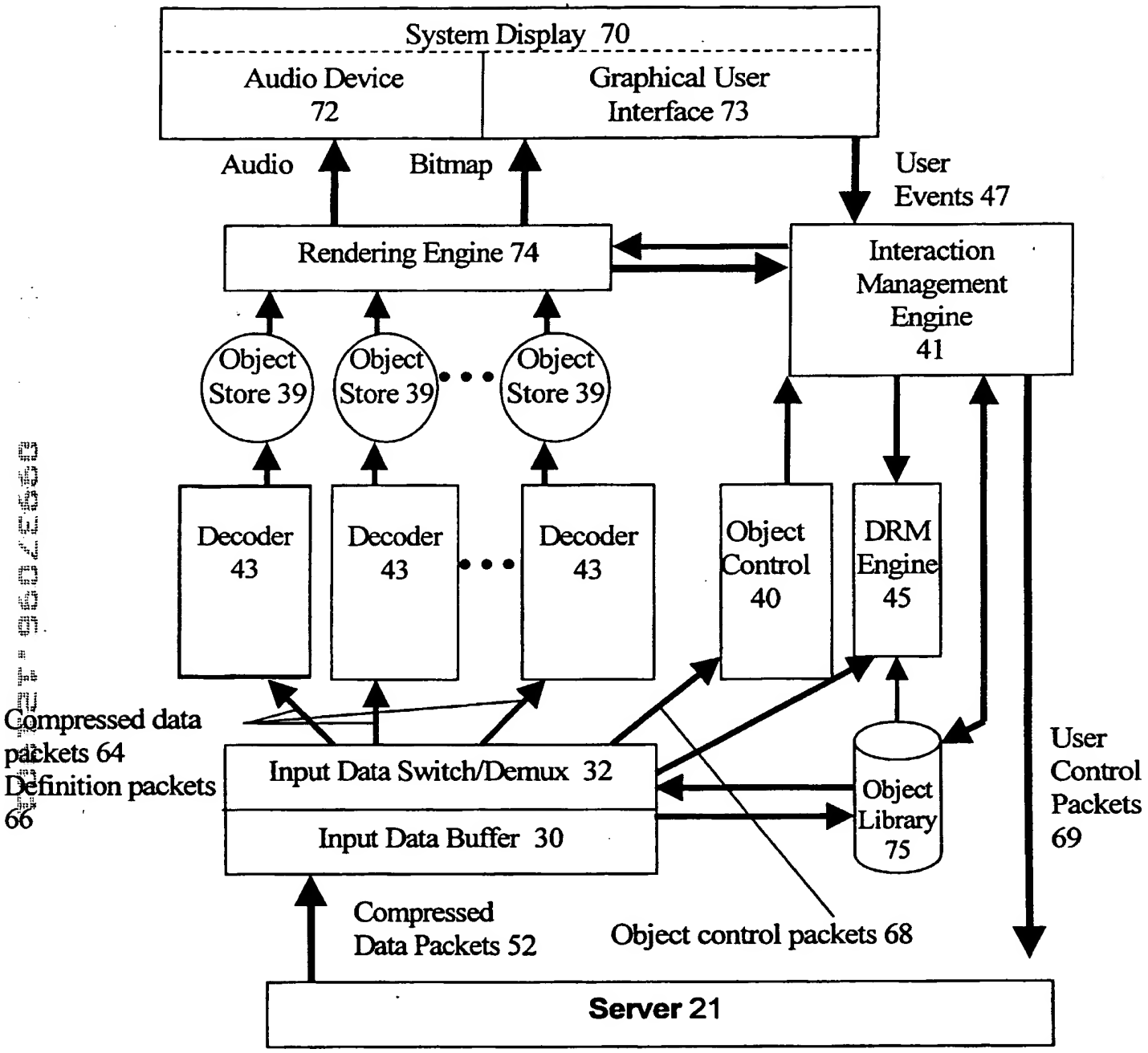
System Display 70
Audio Device 72
Graphical User Interface 73
Audio
Bitmap
User Events 47
Rendering Engine 74
Interaction Management Engine 41
Object Store 39
Object Store 39
Object Store 39
Decoder 43
Decoder 43
Decoder 43
Object Control 40
DRM Engine 45
Compressed data packets 64
Definition packets 66
Input Data Switch/Demux 32
Input Data Buffer 30
Object Library 75
User Control Packets 69
Compressed Data Packets 52
Object control packets 68
Server 21

Figure 8

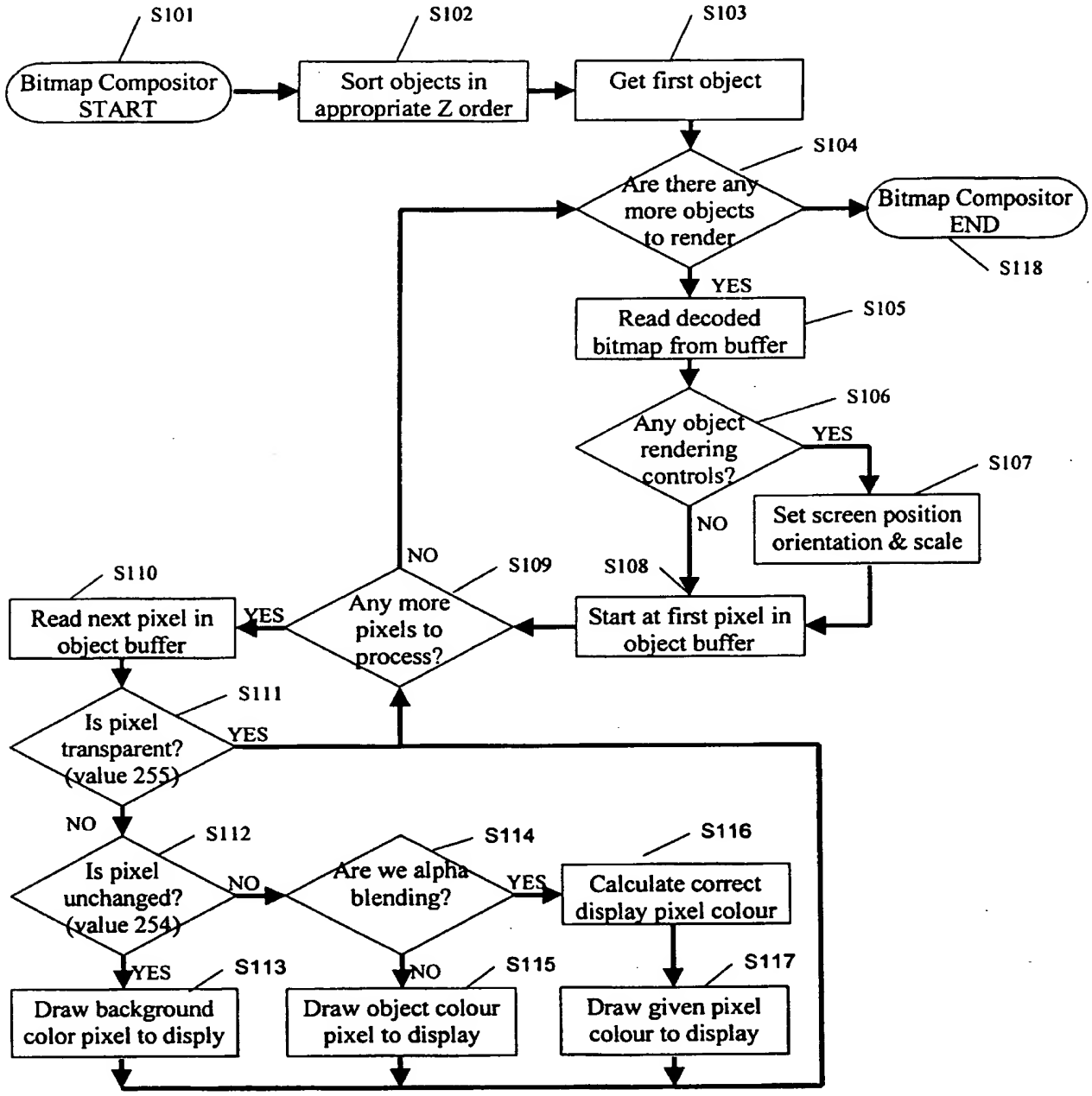
S101
Bitmap Compositor START
S102
Sort objects in appropriate Z order
S103
Get first object
S104
Are there any more objects to render
Bitmap Compositor END
S118
YES
S105
Read decoded bitmap from buffer
S106
Any object rendering controls?
YES
S107
Set screen position orientation & scale
NO
S108
Start at first pixel in object buffer
NO
S109
Any more pixels to process?
YES
S110
Read next pixel in object buffer
S111
Is pixel transparent? (value 255)
YES
NO
S112
Is pixel unchanged? (value 254)
NO
S114
Are we alpha blending?
YES
S116
Calculate correct display pixel colour
YES
S113
Draw background color pixel to disply
NO
S115
Draw object colour pixel to display
S117
Draw given pixel colour to display

Figure 9

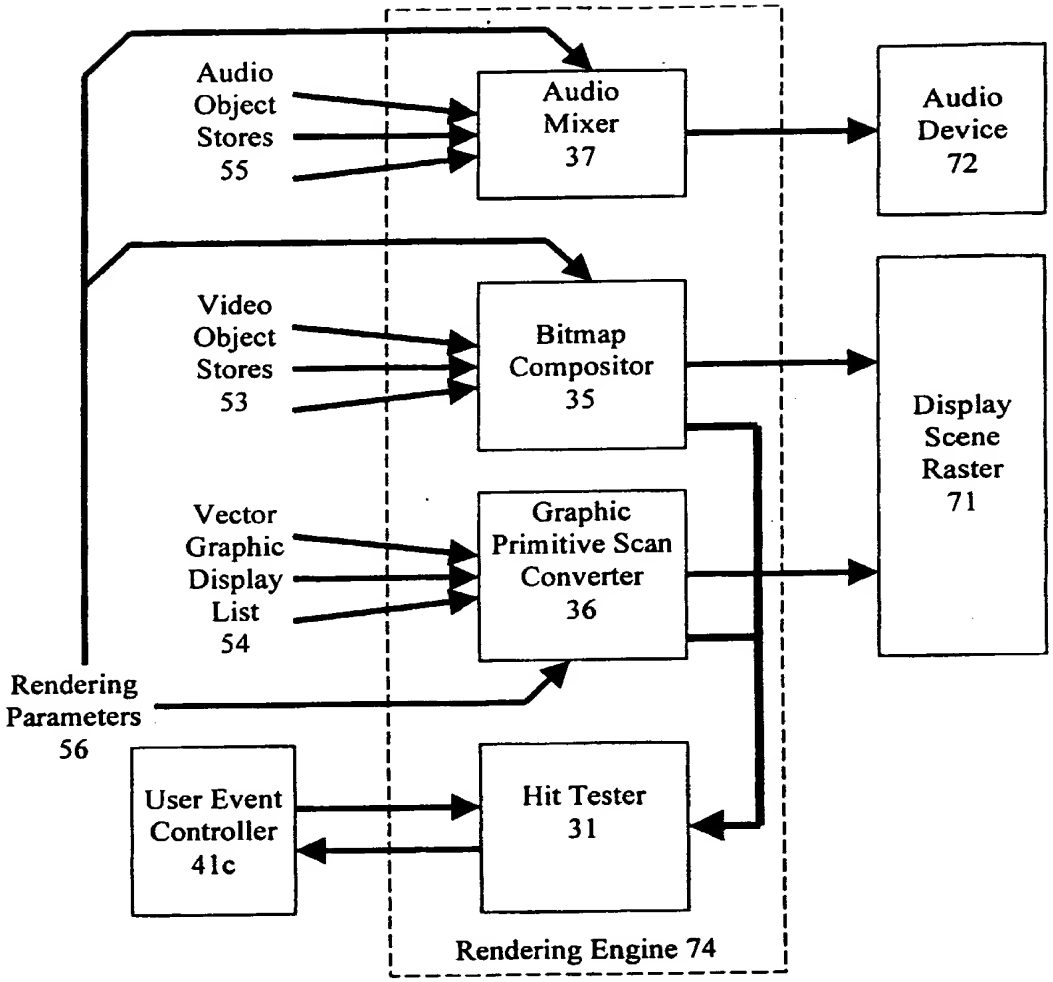
Audio
Object
Stores
55
Audio
Mixer
37
Audio
Device
72
Video
Object
Stores
53
Bitmap
Compositor
35
Display
Scene
Raster
71
Vector
Graphic
Display
List
54
Graphic
Primitive Scan
Converter
36
Rendering
Parameters
56
User Event
Controller
41c
Hit Tester
31
Rendering Engine 74

Figure 10

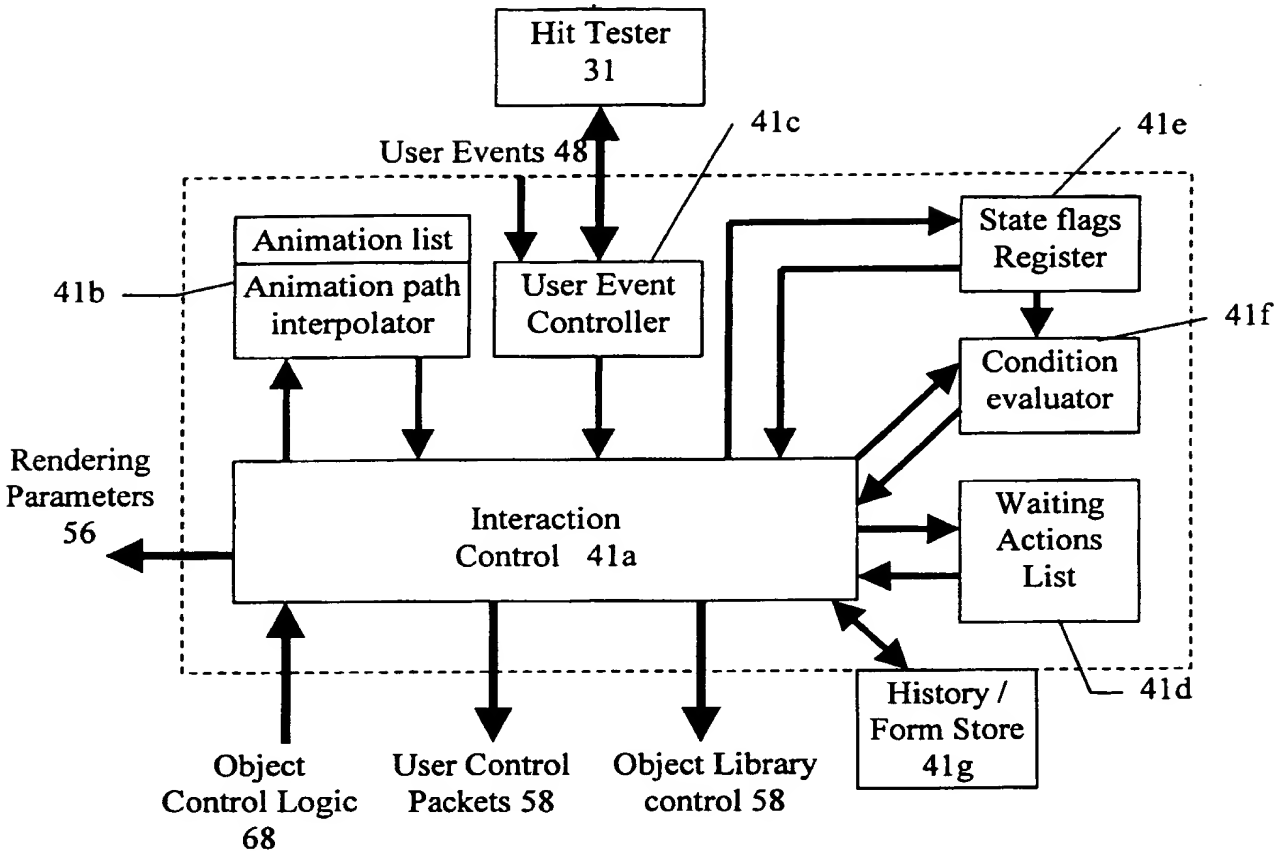
Hit Tester
31
User Events 48
41c
41e
Animation list
Animation path
interpolator
41b
User Event
Controller
State flags
Register
41f
Condition
evaluator
Rendering
Parameters
56
Interaction
Control 41a
Waiting
Actions
List
41d
History /
Form Store
41g
Object
Control Logic
68
User Control
Packets 58
Object Library
control 58

Figure 11

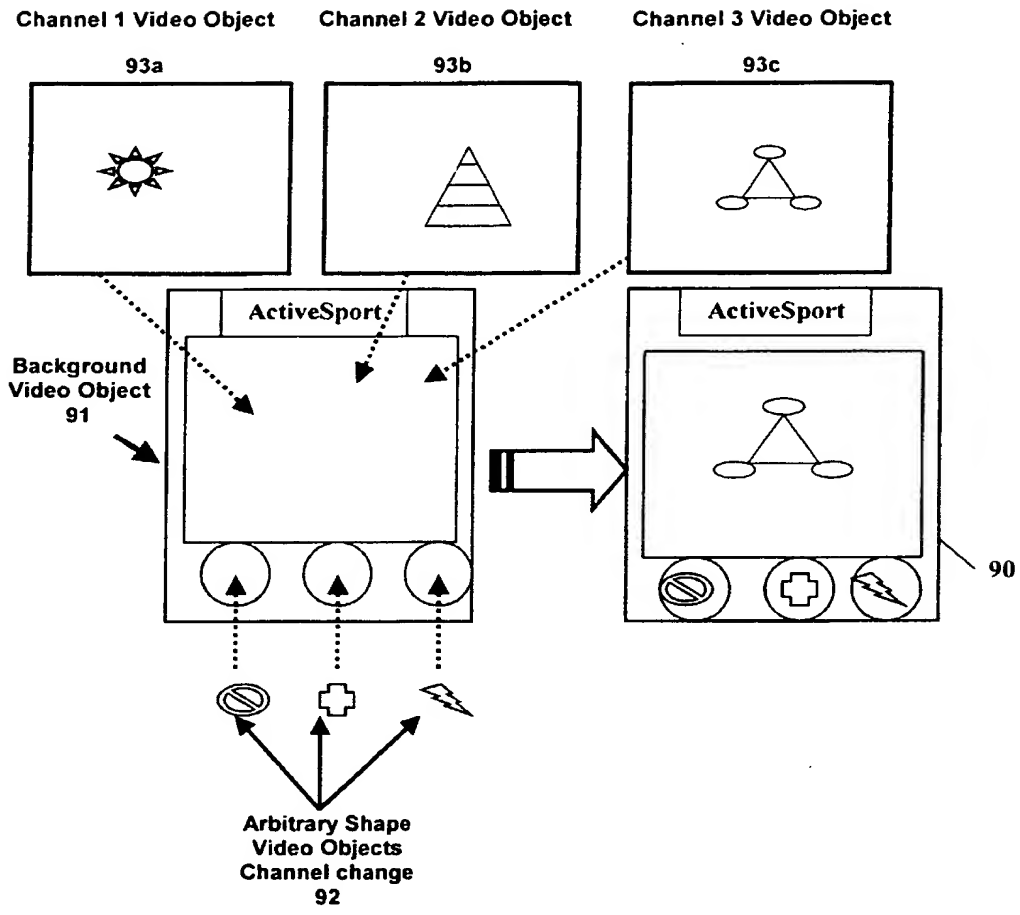
Channel 1 Video Object
Channel 2 Video Object
Channel 3 Video Object
93a
93b
93c
ActiveSport
ActiveSport
Background Video Object 91
90
Arbitrary Shape Video Objects Channel change 92

Figure 12

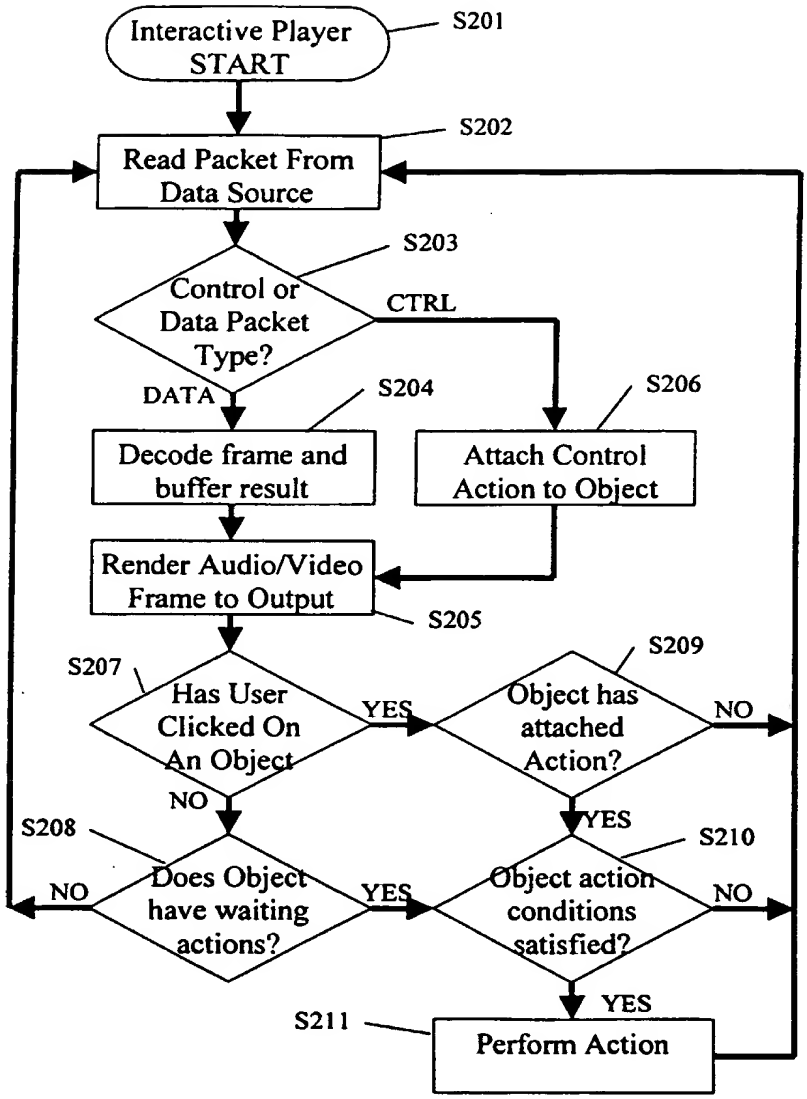
Interactive Player START
S201
Read Packet From Data Source
S202
Control or Data Packet Type?
S203
CTRL
DATA
Decode frame and buffer result
S204
Attach Control Action to Object
S206
Render Audio/Video Frame to Output
S205
Has User Clicked On An Object
S207
YES
Object has attached Action?
S209
NO
NO
YES
Does Object have waiting actions?
S208
NO
YES
Object action conditions satisfied?
S210
NO
YES
Perform Action
S211

Figure 13

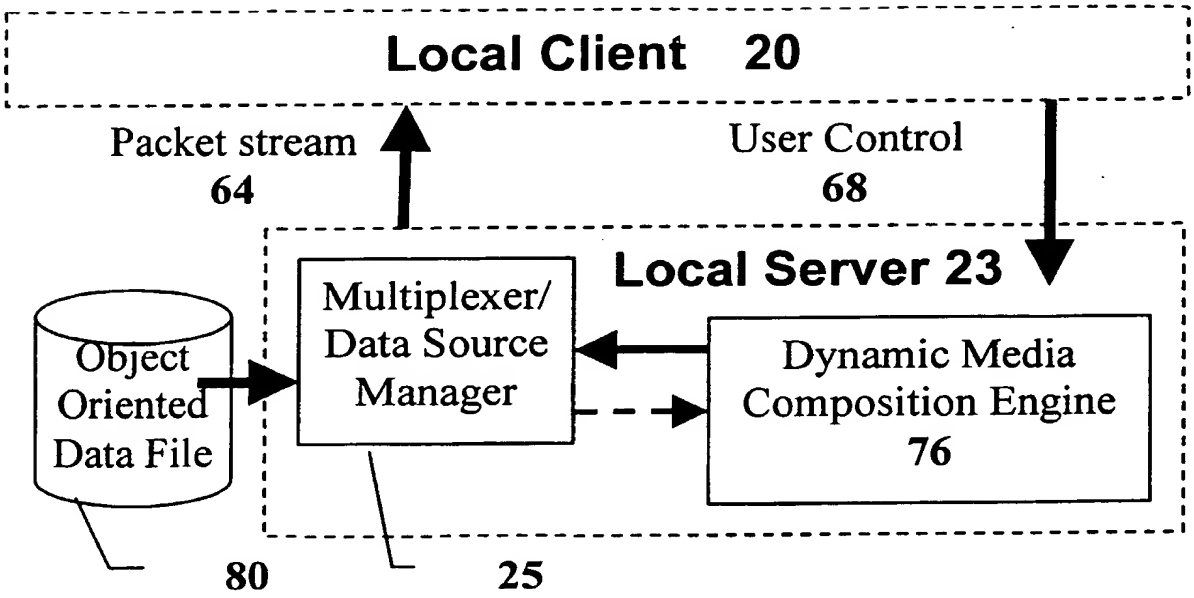
Local Client 20
Packet stream
64
User Control
68
Local Server 23
Multiplexer/
Data Source
Manager
Object
Oriented
Data File
Dynamic Media
Composition Engine
76
80
25

Figure 14

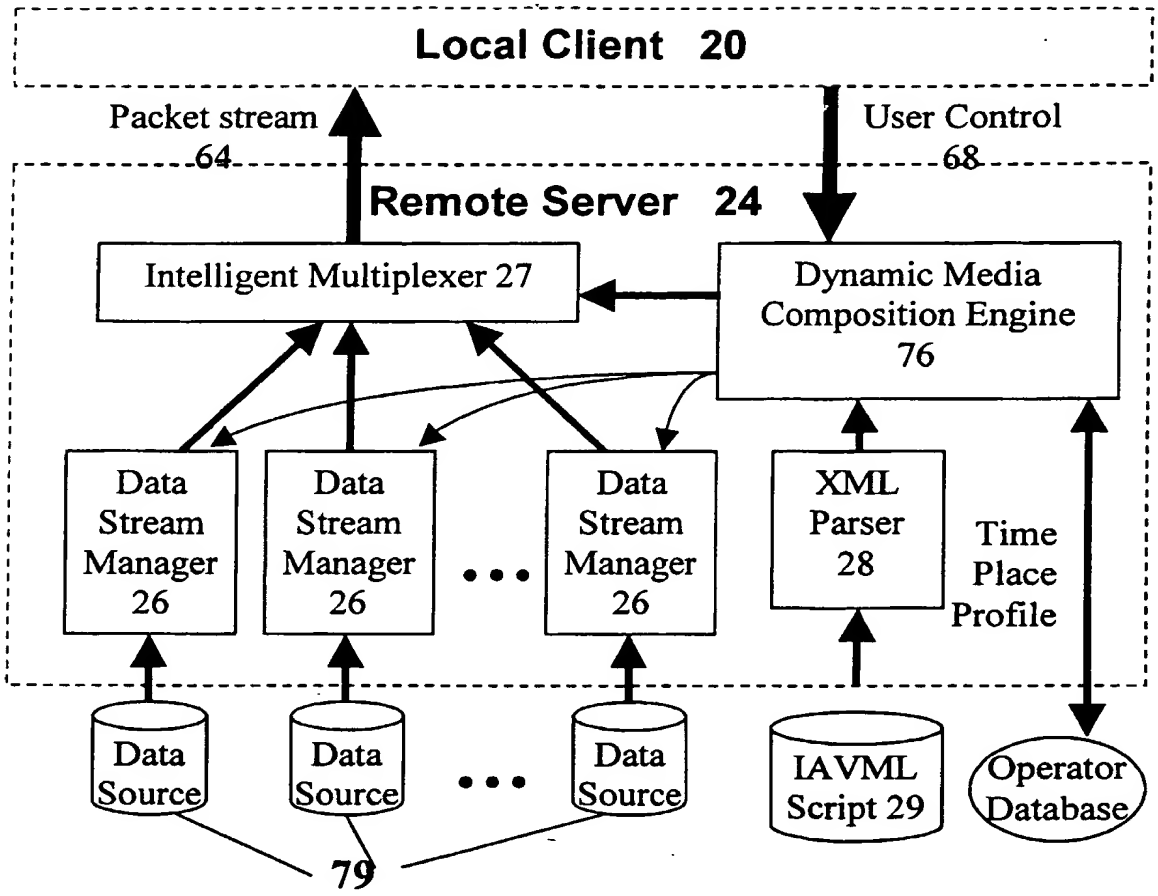
Local Client 20
Packet stream 64
User Control 68
Remote Server 24
Intelligent Multiplexer 27
Dynamic Media Composition Engine 76
Data Stream Manager 26
Data Stream Manager 26
Data Stream Manager 26
XML Parser 28
Time Place Profile
Data Source
Data Source
Data Source
IAVML Script 29
Operator Database
79

Figure 15

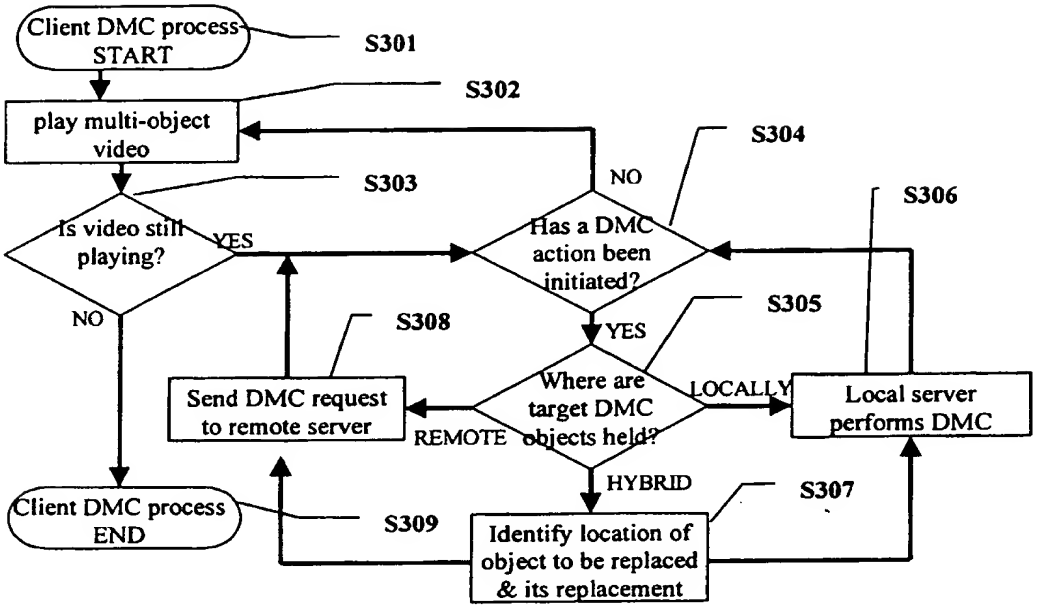
Client DMC process START
S301
S302
play multi-object video
S303
Is video still playing?
YES
NO
S304
Has a DMC action been initiated?
NO
YES
S306
S305
Where are target DMC objects held?
LOCALLY
REMOTE
HYBRID
S308
Send DMC request to remote server
Local server performs DMC
S307
Identify location of object to be replaced & its replacement
Client DMC process END
S309

Figure 16

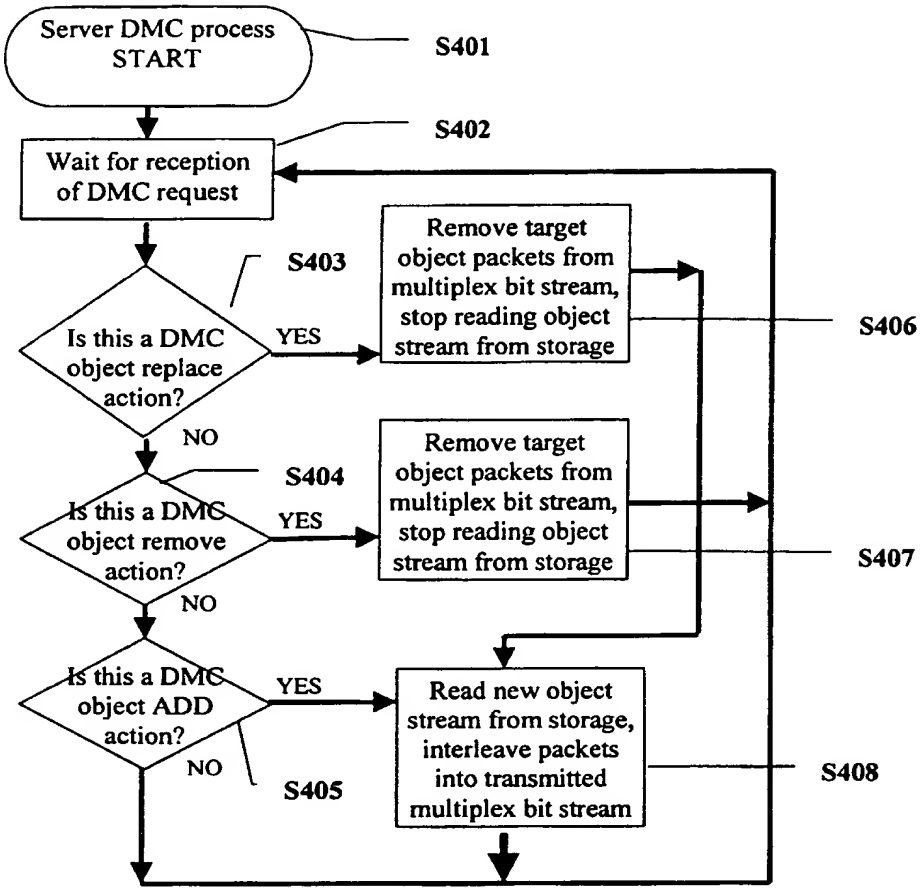
Server DMC process START
S401
S402
Wait for reception of DMC request
S403
Is this a DMC object replace action?
YES
Remove target object packets from multiplex bit stream, stop reading object stream from storage
S406
NO
S404
Is this a DMC object remove action?
YES
Remove target object packets from multiplex bit stream, stop reading object stream from storage
S407
NO
Is this a DMC object ADD action?
YES
Read new object stream from storage, interleave packets into transmitted multiplex bit stream
S408
NO
S405

Figure 17

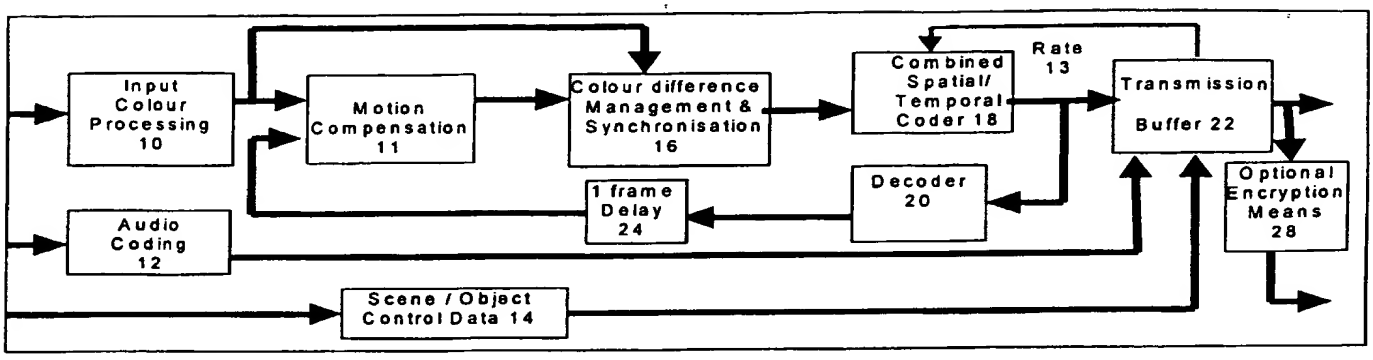
Input Colour Processing 10
Motion Compensation 11
Colour difference Management & Synchronisation 16
Combined Spatial/ Temporal Coder 18
Rate 13
Transmission Buffer 22
Optional Encryption Means 28
Decoder 20
1 frame Delay 24
Audio Coding 12
Scene / Object Control Data 14

Figure 18

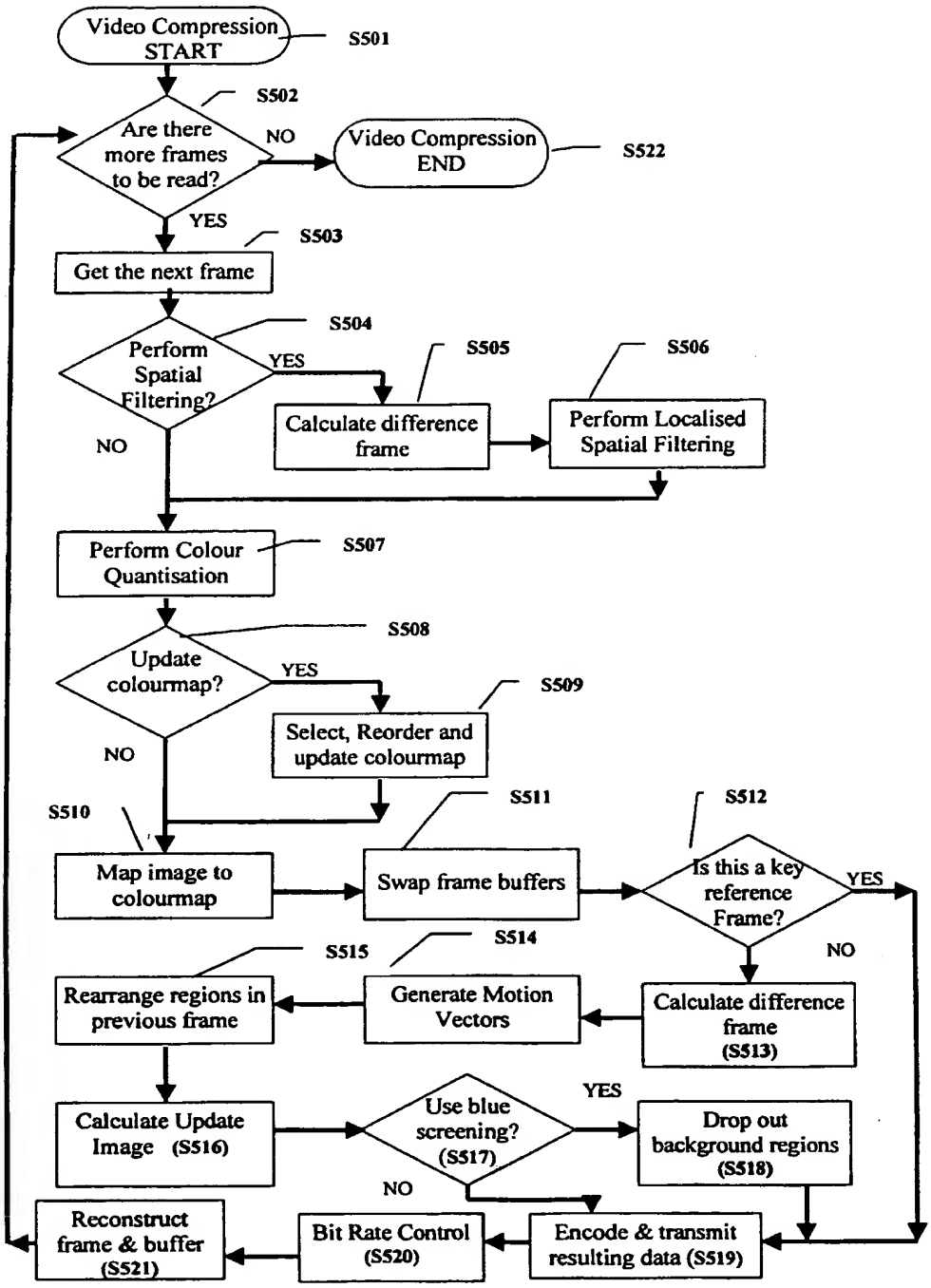
Video Compression START
S501
S502
Are there more frames to be read?
NO
Video Compression END
S522
YES
S503
Get the next frame
S504
Perform Spatial Filtering?
YES
S505
Calculate difference frame
S506
Perform Localised Spatial Filtering
NO
Perform Colour Quantisation
S507
S508
Update colourmap?
YES
S509
Select, Reorder and update colourmap
NO
S510
Map image to colourmap
S511
Swap frame buffers
S512
Is this a key reference Frame?
YES
NO
S514
S515
Rearrange regions in previous frame
Generate Motion Vectors
Calculate difference frame (S513)
YES
Calculate Update Image (S516)
Use blue screening? (S517)
Drop out background regions (S518)
NO
Reconstruct frame & buffer (S521)
Bit Rate Control (S520)
Encode & transmit resulting data (S519)

Figure 19

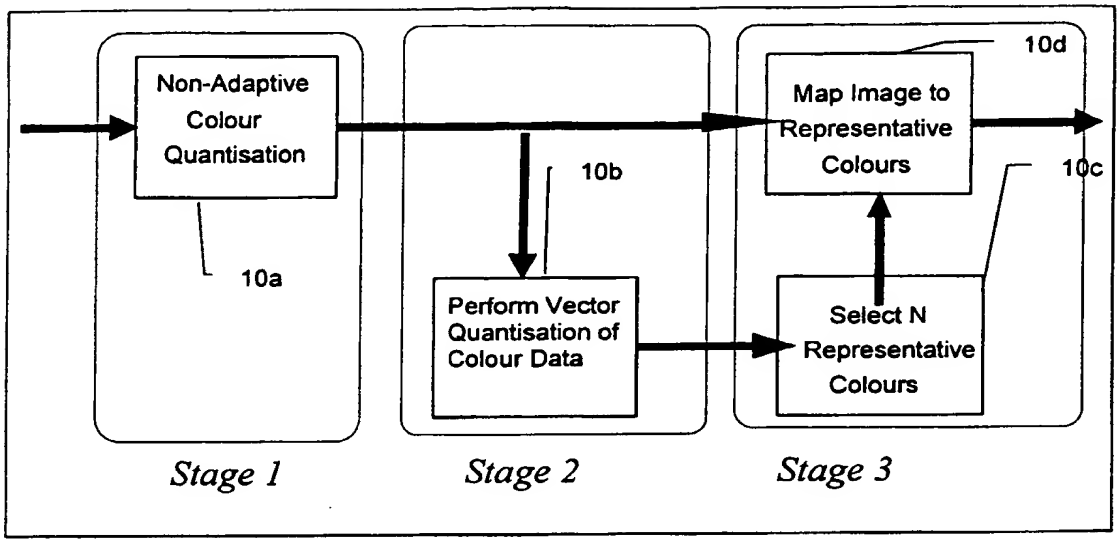

Figure 20

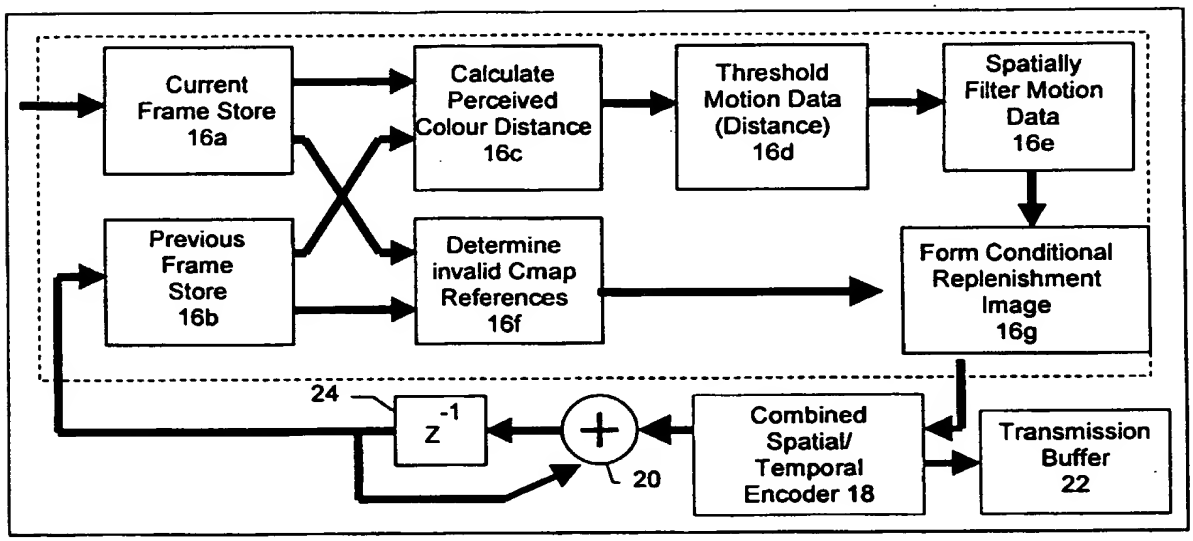

Figure 21

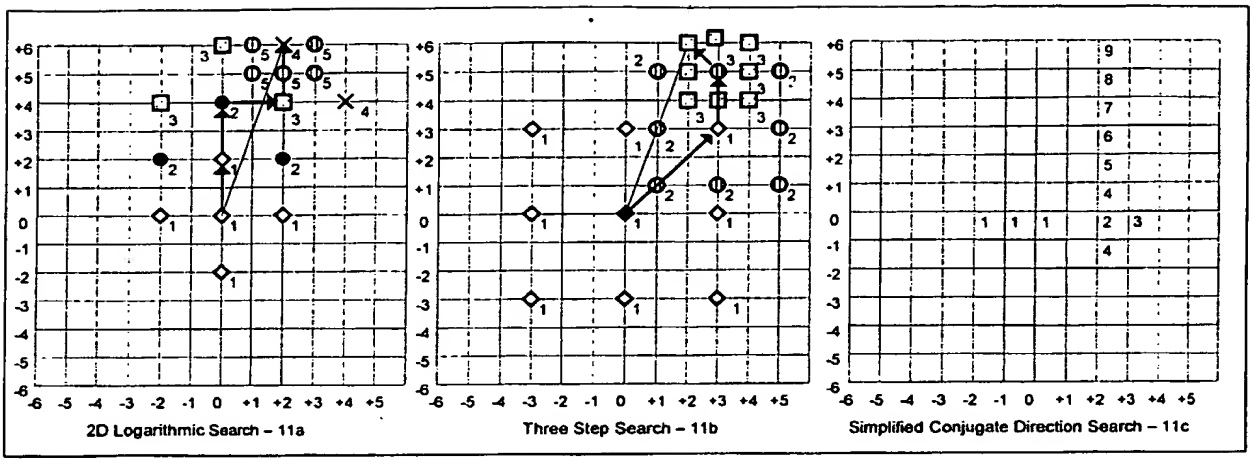

Figure 22

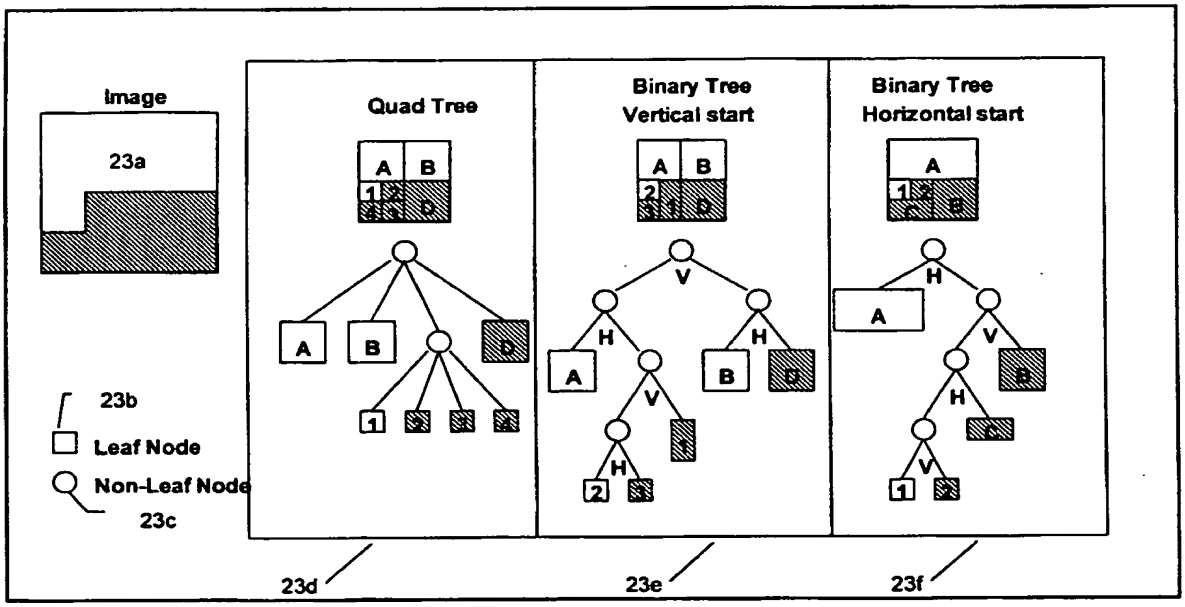

Figure 23

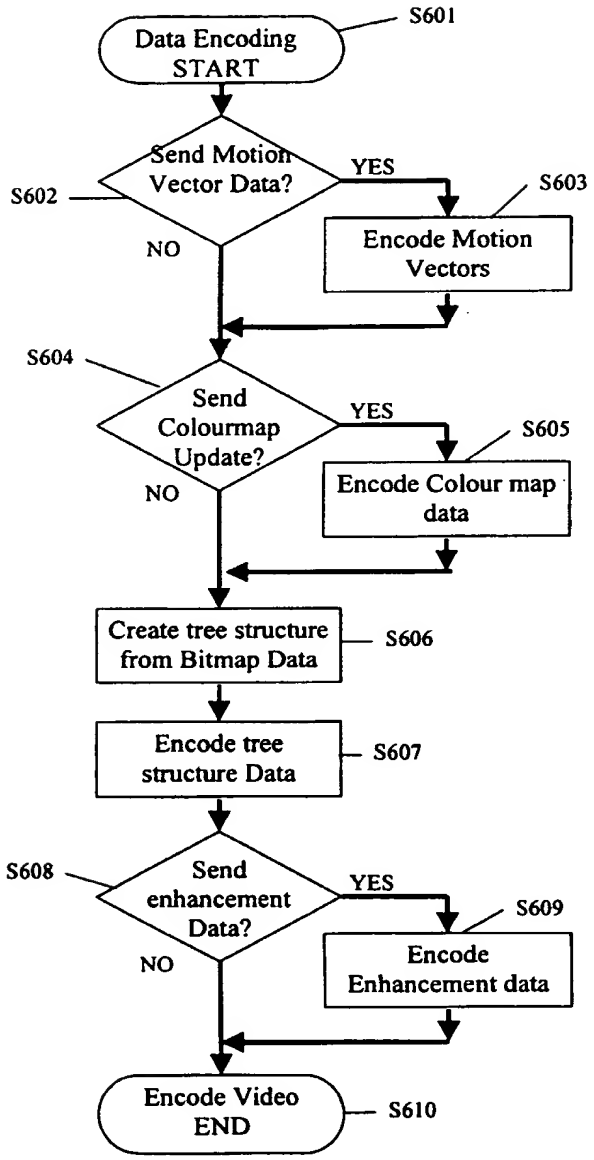
Data Encoding START
S601
Send Motion Vector Data?
S602
YES
S603
Encode Motion Vectors
NO
S604
Send Colourmap Update?
YES
S605
Encode Colour map data
NO
Create tree structure from Bitmap Data
S606
Encode tree structure Data
S607
S608
Send enhancement Data?
YES
S609
Encode Enhancement data
NO
Encode Video END
S610

Figure 24

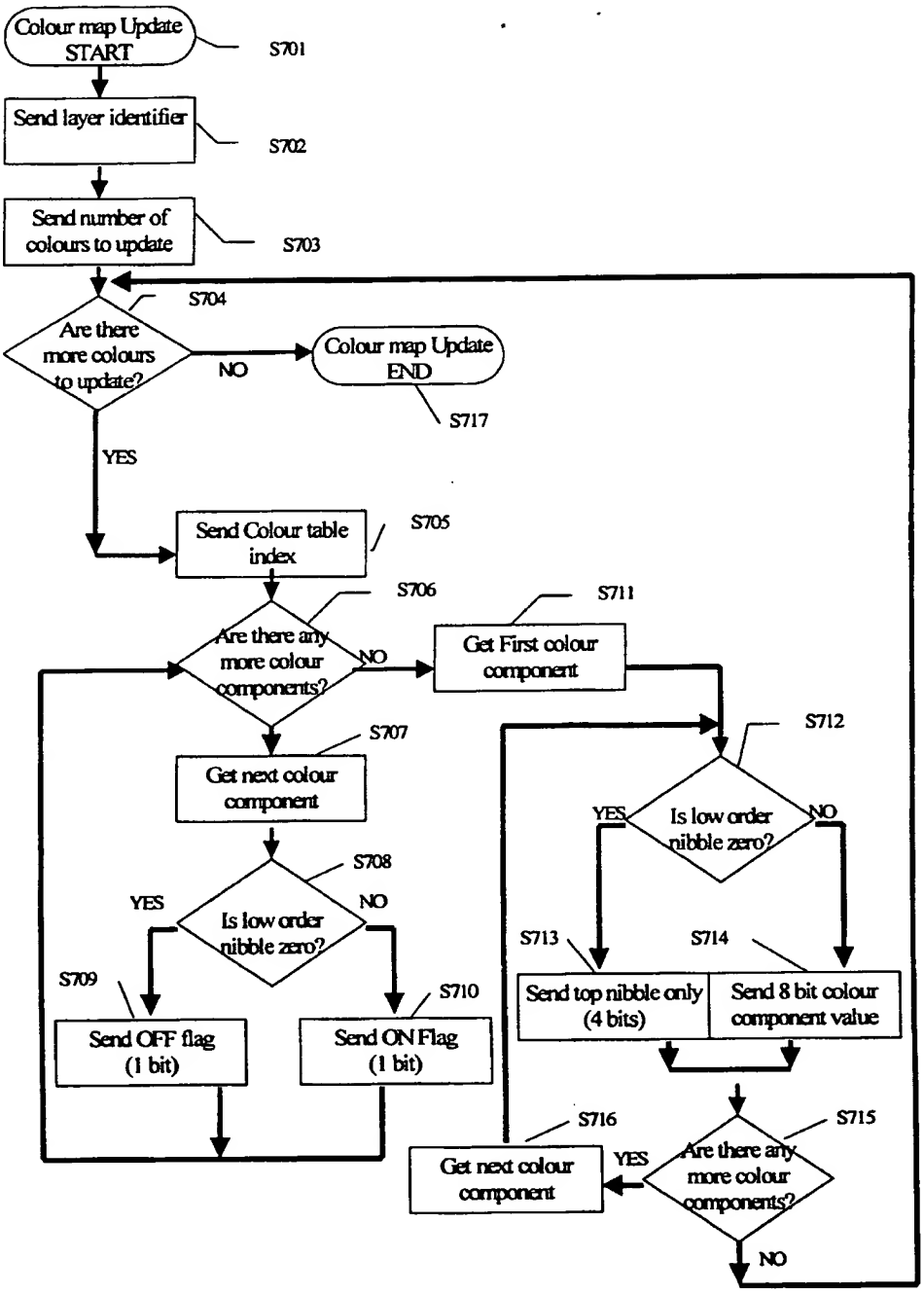
Colour map Update START
S701
Send layer identifier
S702
Send number of colours to update
S703
S704
Are there more colours to update?
NO
Colour map Update END
S717
YES
Send Colour table index
S705
S706
Are there any more colour components?
NO
S711
Get First colour component
S707
Get next colour component
S708
YES
Is low order nibble zero?
NO
S709
Send OFF flag (1 bit)
S710
Send ON Flag (1 bit)
S712
YES
Is low order nibble zero?
NO
S713
S714
Send top nibble only (4 bits)
Send 8 bit colour component value
S716
S715
Get next colour component
YES
Are there any more colour components?
NO

Figure 25

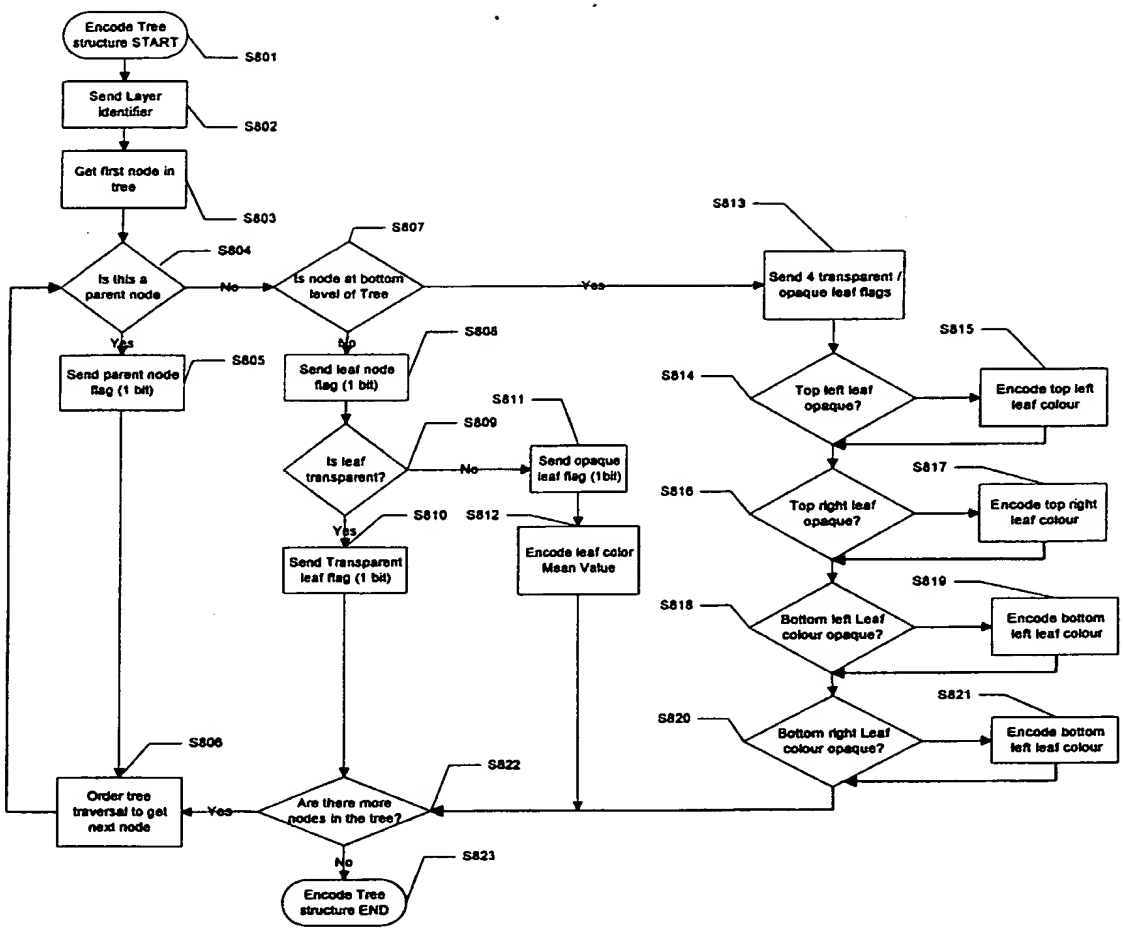
Encode Tree structure START
S801
Send Layer identifier
S802
Get first node in tree
S803
Is this a parent node
S804
No
Yes
Send parent node flag (1 bit)
S805
Order tree traversal to get next node
S806
Is node at bottom level of Tree
S807
No
Send leaf node flag (1 bit)
S808
Is leaf transparent?
S809
No
Yes
Send Transparent leaf flag (1 bit)
S810
Send opaque leaf flag (1bit)
S811
Encode leaf color Mean Value
S812
Are there more nodes in the tree?
S822
Yes
No
Encode Tree structure END
S823
Yes
Send 4 transparent / opaque leaf flags
S813
Top left leaf opaque?
S814
Encode top left leaf colour
S815
Top right leaf opaque?
S816
Encode top right leaf colour
S817
Bottom left Leaf colour opaque?
S818
Encode bottom left leaf colour
S819
Bottom right Leaf colour opaque?
S820
Encode bottom left leaf colour
S821

Figure 26

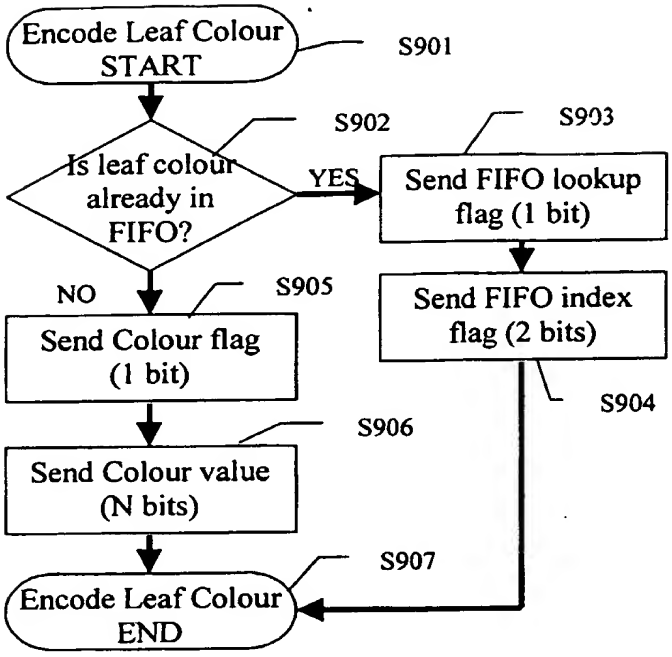
Encode Leaf Colour START
S901
S902
Is leaf colour already in FIFO?
YES
S903
Send FIFO lookup flag (1 bit)
Send FIFO index flag (2 bits)
S904
NO
S905
Send Colour flag (1 bit)
S906
Send Colour value (N bits)
S907
Encode Leaf Colour END

Figure 27

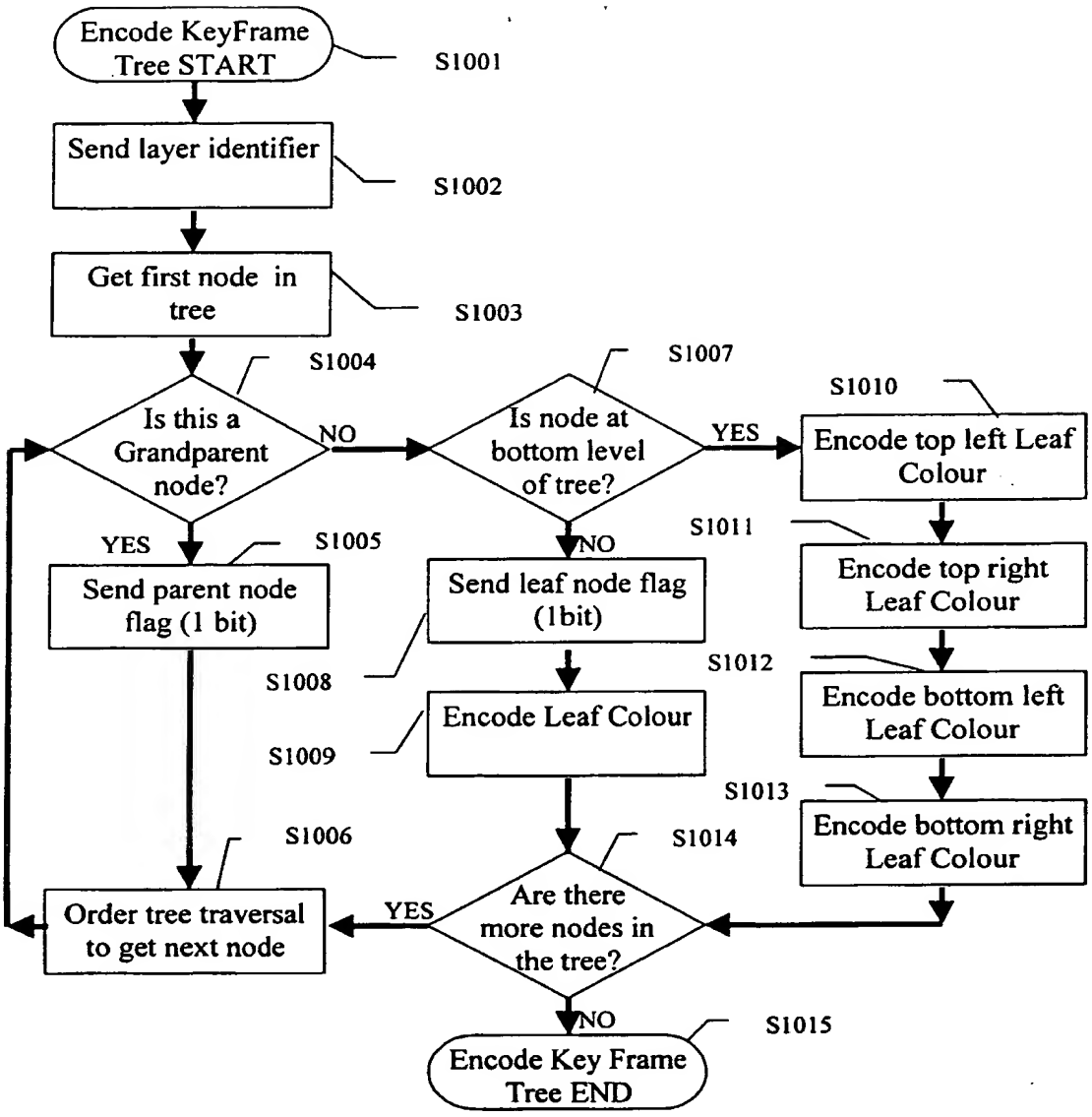
Encode KeyFrame Tree START
S1001
Send layer identifier
S1002
Get first node in tree
S1003
S1004
Is this a Grandparent node?
NO
YES
S1005
Send parent node flag (1 bit)
S1006
Order tree traversal to get next node
S1007
Is node at bottom level of tree?
YES
NO
Send leaf node flag (1bit)
S1008
Encode Leaf Colour
S1009
S1010
Encode top left Leaf Colour
S1011
Encode top right Leaf Colour
S1012
Encode bottom left Leaf Colour
S1013
Encode bottom right Leaf Colour
S1014
Are there more nodes in the tree?
YES
NO
S1015
Encode Key Frame Tree END

Figure 28

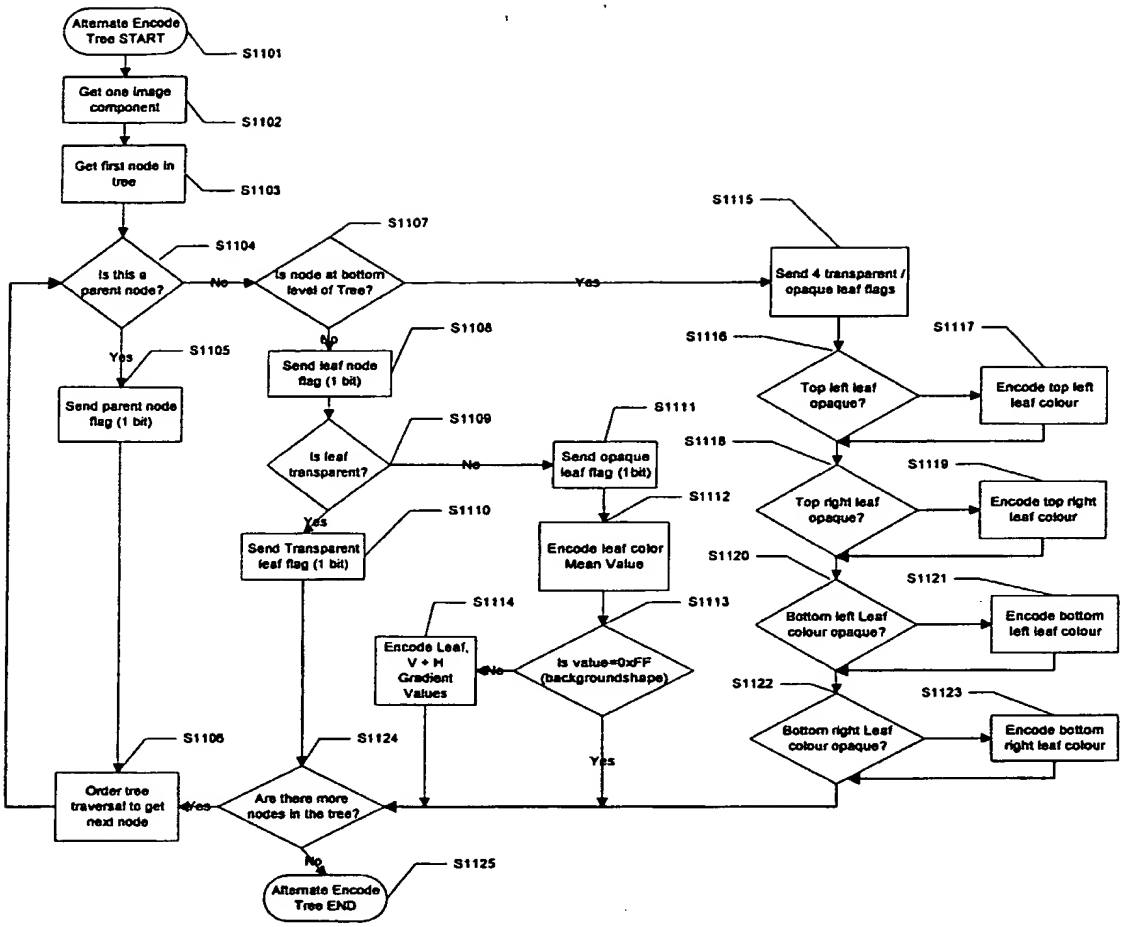
Alternate Encode Tree START
S1101
Get one image component
S1102
Get first node in tree
S1103
Is this a parent node?
S1104
No
Is node at bottom level of Tree?
S1107
Yes
Send 4 transparent / opaque leaf flags
S1115
Yes
Send parent node flag (1 bit)
S1105
No
Send leaf node flag (1 bit)
S1108
Is leaf transparent?
S1109
No
Send opaque leaf flag (1bit)
S1111
Yes
Send Transparent leaf flag (1 bit)
S1110
Encode leaf color Mean Value
S1112
Is value=0xFF (backgroundshape)
S1113
No
Encode Leaf, V + H Gradient Values
S1114
Yes
Top left leaf opaque?
S1116
Encode top left leaf colour
S1117
Top right leaf opaque?
S1118
Encode top right leaf colour
S1119
Bottom left Leaf colour opaque?
S1120
Encode bottom left leaf colour
S1121
Bottom right Leaf colour opaque?
S1122
Encode bottom right leaf colour
S1123
Order tree traversal to get next node
S1106
Are there more nodes in the tree?
S1124
Yes
No
Alternate Encode Tree END
S1125

Figure 29

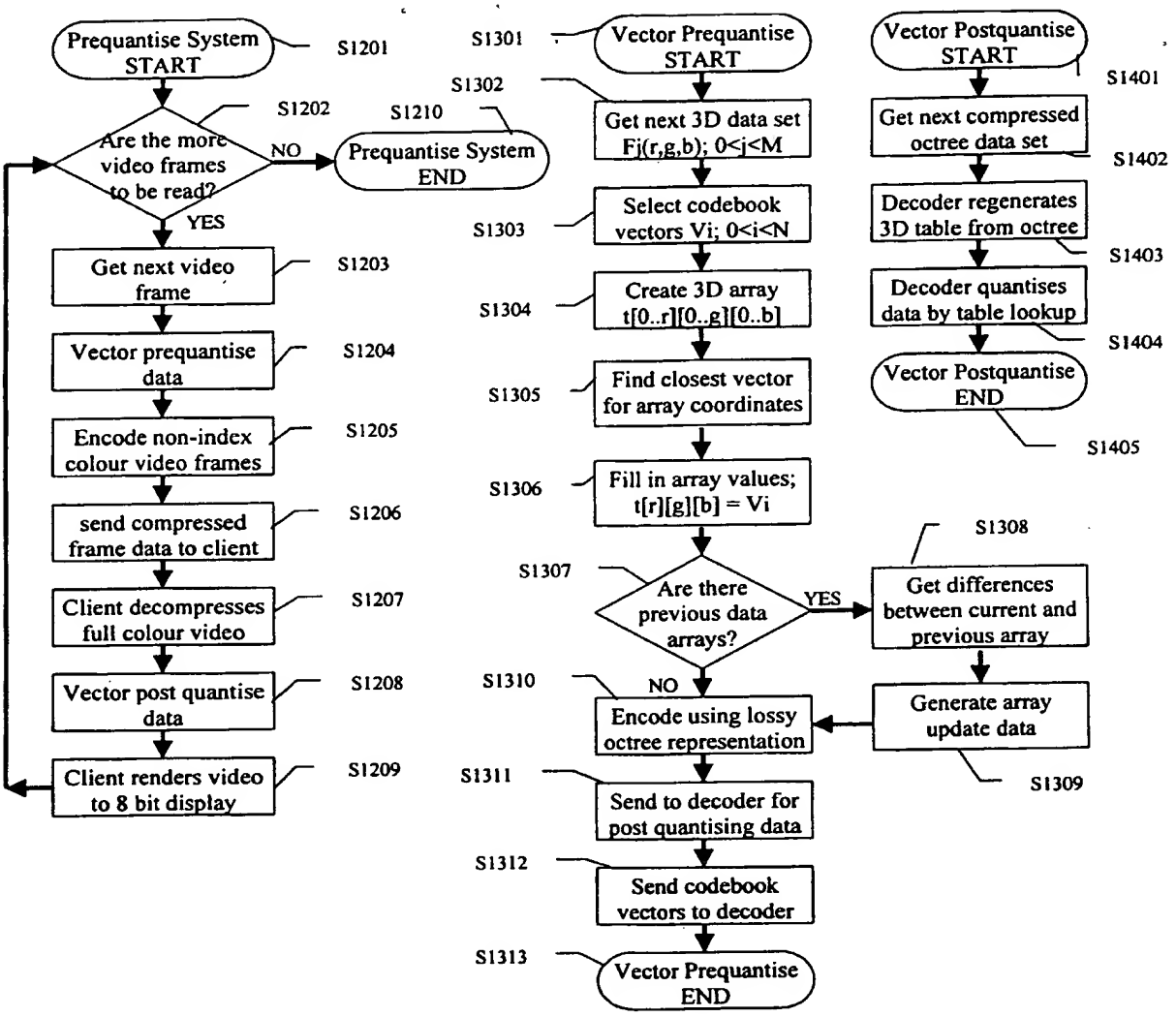
Prequantise System START
S1201
Are the more video frames to be read?
S1202
NO
Prequantise System END
S1210
YES
Get next video frame
S1203
Vector prequantise data
S1204
Encode non-index colour video frames
S1205
send compressed frame data to client
S1206
Client decompresses full colour video
S1207
Vector post quantise data
S1208
Client renders video to 8 bit display
S1209
S1301
Vector Prequantise START
S1302
Get next 3D data set Fj(r,g,b); 0<j<M
S1303
Select codebook vectors Vi; 0<i<N
S1304
Create 3D array t[0..r][0..g][0..b]
S1305
Find closest vector for array coordinates
S1306
Fill in array values; t[r][g][b] = Vi
S1307
Are there previous data arrays?
YES
S1308
Get differences between current and previous array
S1309
NO
S1310
Encode using lossy octree representation
Generate array update data
S1311
Send to decoder for post quantising data
S1312
Send codebook vectors to decoder
S1313
Vector Prequantise END
Vector Postquantise START
S1401
Get next compressed octree data set
S1402
Decoder regenerates 3D table from octree
S1403
Decoder quantises data by table lookup
S1404
Vector Postquantise END
S1405

Figure 30

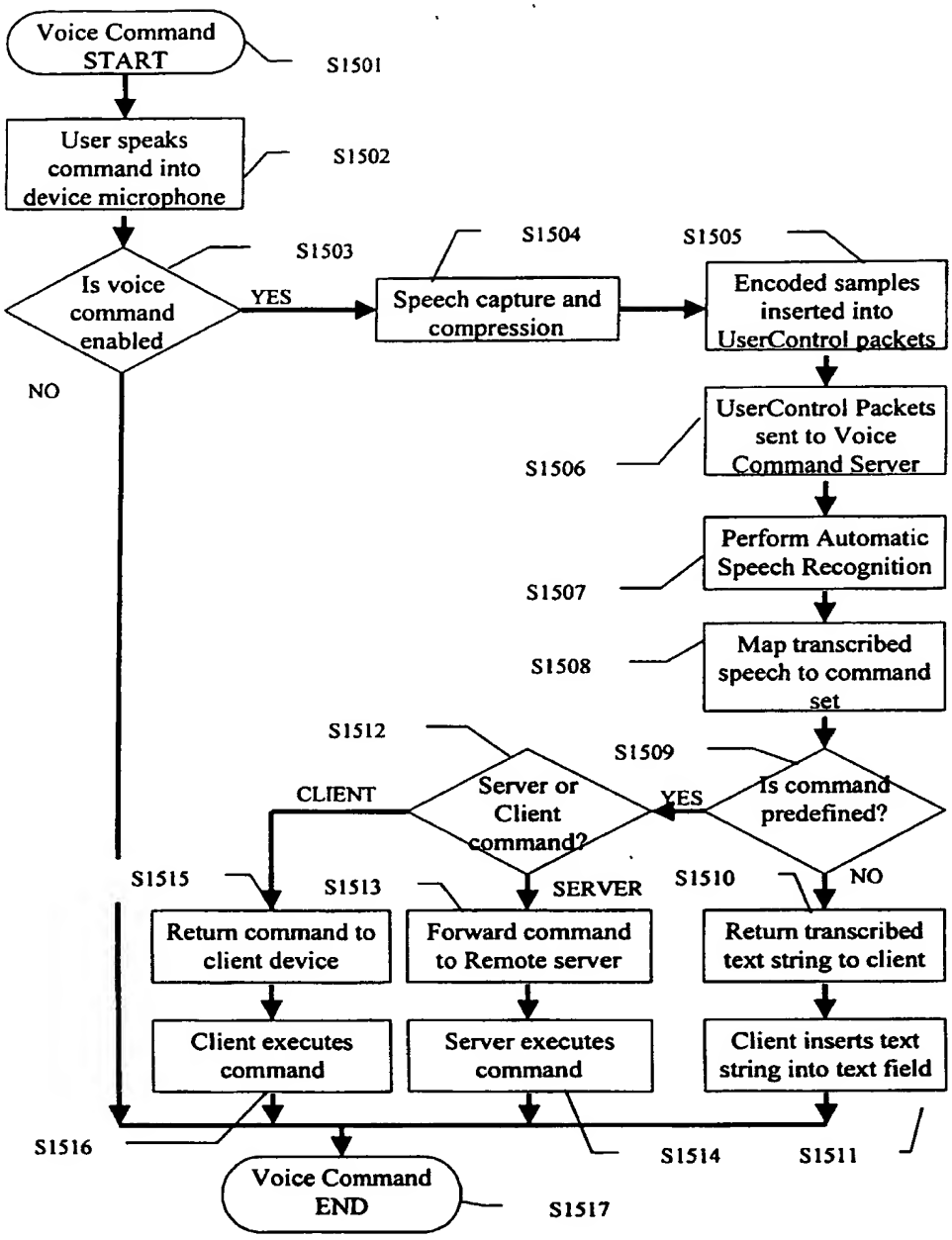
Voice Command START
S1501
User speaks command into device microphone
S1502
S1503
Is voice command enabled
YES
NO
S1504
Speech capture and compression
S1505
Encoded samples inserted into UserControl packets
UserControl Packets sent to Voice Command Server
S1506
Perform Automatic Speech Recognition
S1507
Map transcribed speech to command set
S1508
S1509
Is command predefined?
YES
NO
S1512
Server or Client command?
CLIENT
SERVER
S1515
Return command to client device
S1513
Forward command to Remote server
S1510
Return transcribed text string to client
Client executes command
Server executes command
Client inserts text string into text field
S1516
S1514
S1511
Voice Command END
S1517

Figure 31

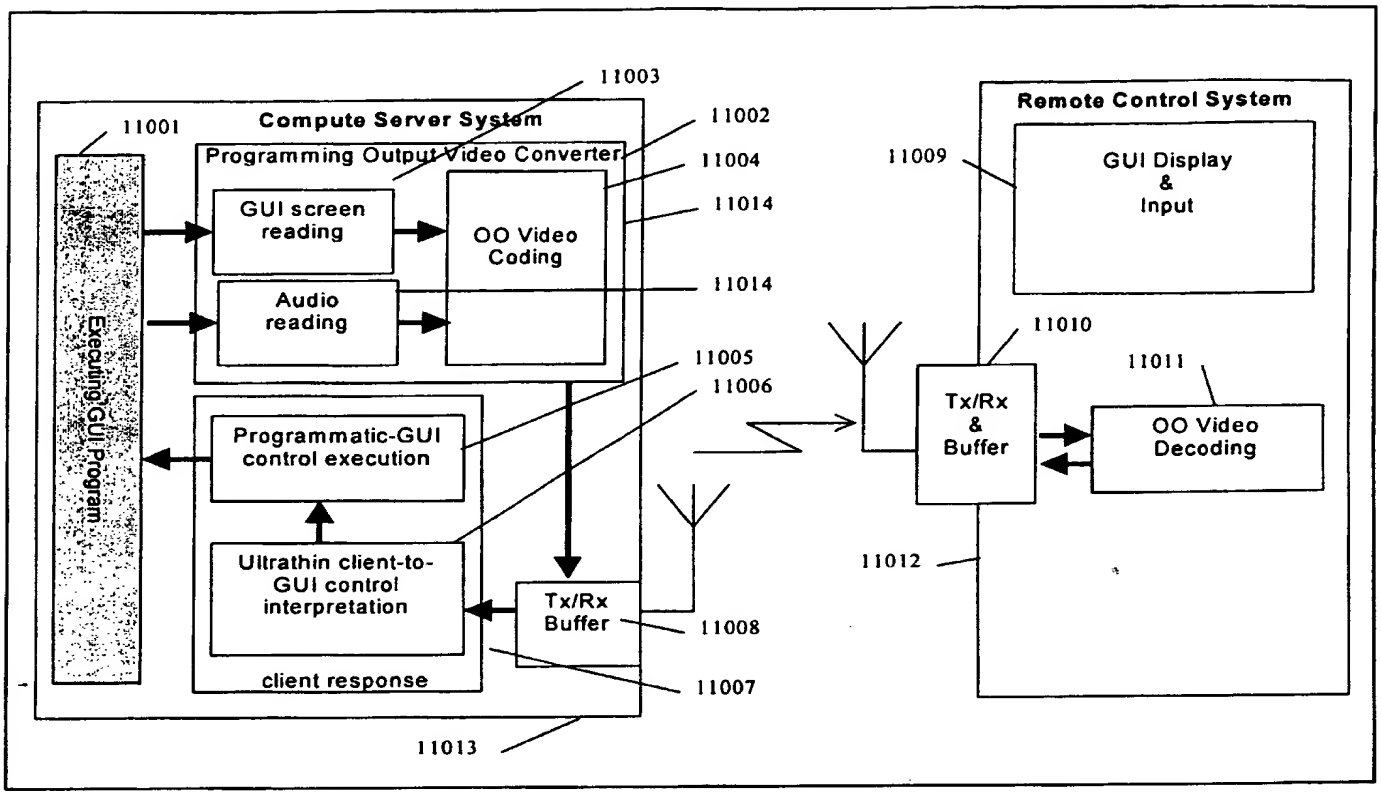
11003
11001
Compute Server System
11002
Programming Output Video Converter
11004
GUI screen reading
OO Video Coding
11014
Audio reading
11014
Executing GUI Program
11005
11006
Programmatic-GUI control execution
Ultrathin client-to-GUI control interpretation
Tx/Rx Buffer
11008
client response
11007
11013
Remote Control System
11009
GUI Display & Input
11010
11011
Tx/Rx & Buffer
OO Video Decoding
11012

Figure 32

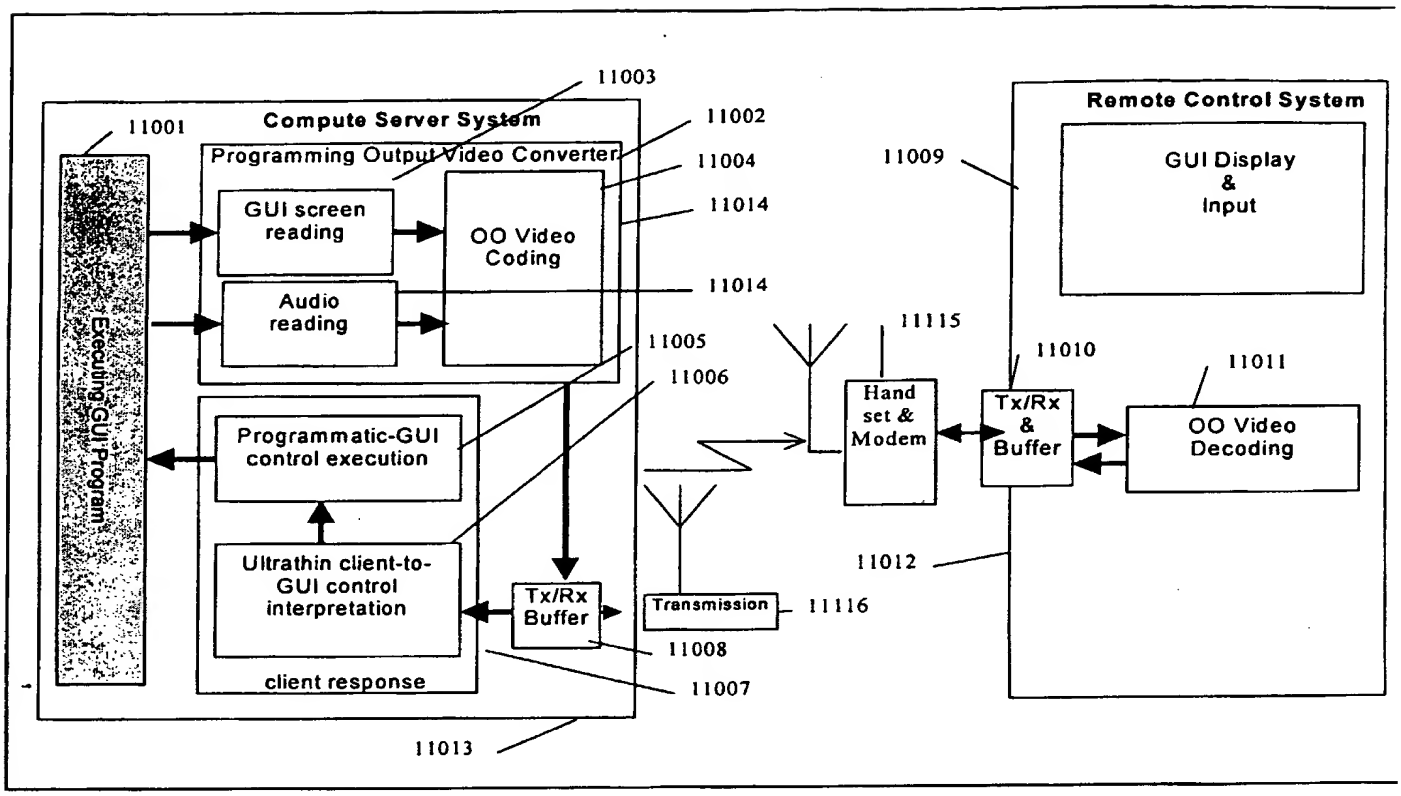
11003
11001
Compute Server System
11002
Programming Output Video Converter
11004
11009
Remote Control System
GUI Display & Input
GUI screen reading
OO Video Coding
11014
Audio reading
11014
11115
11005
11010
11011
11006
Executing GUI Program
Programmatic-GUI control execution
Hand set & Modem
Tx/Rx & Buffer
OO Video Decoding
Ultrathin client-to-GUI control interpretation
Tx/Rx Buffer
Transmission
11116
11012
11008
client response
11007
11013

Figure 33

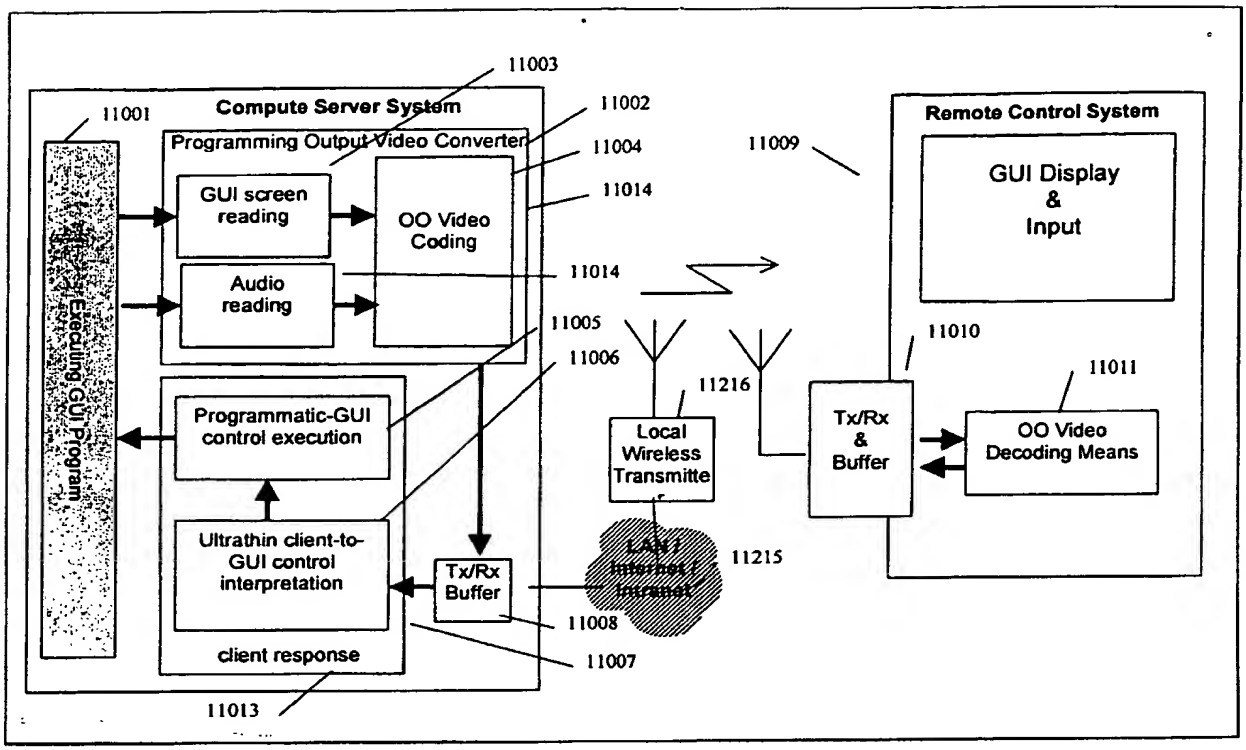
11003
11001
Compute Server System
11002
Programming Output Video Converter
11004
11009
Remote Control System
GUI screen reading
OO Video Coding
11014
GUI Display & Input
Audio reading
11014
11005
11010
11006
11011
11216
Programmatic-GUI control execution
Local Wireless Transmitte
Tx/Rx & Buffer
OO Video Decoding Means
Executing GUI Program
Ultrathin client-to-GUI control interpretation
Tx/Rx Buffer
LAN / Internet / Intranet
11215
11008
client response
11007
11013

Figure 34

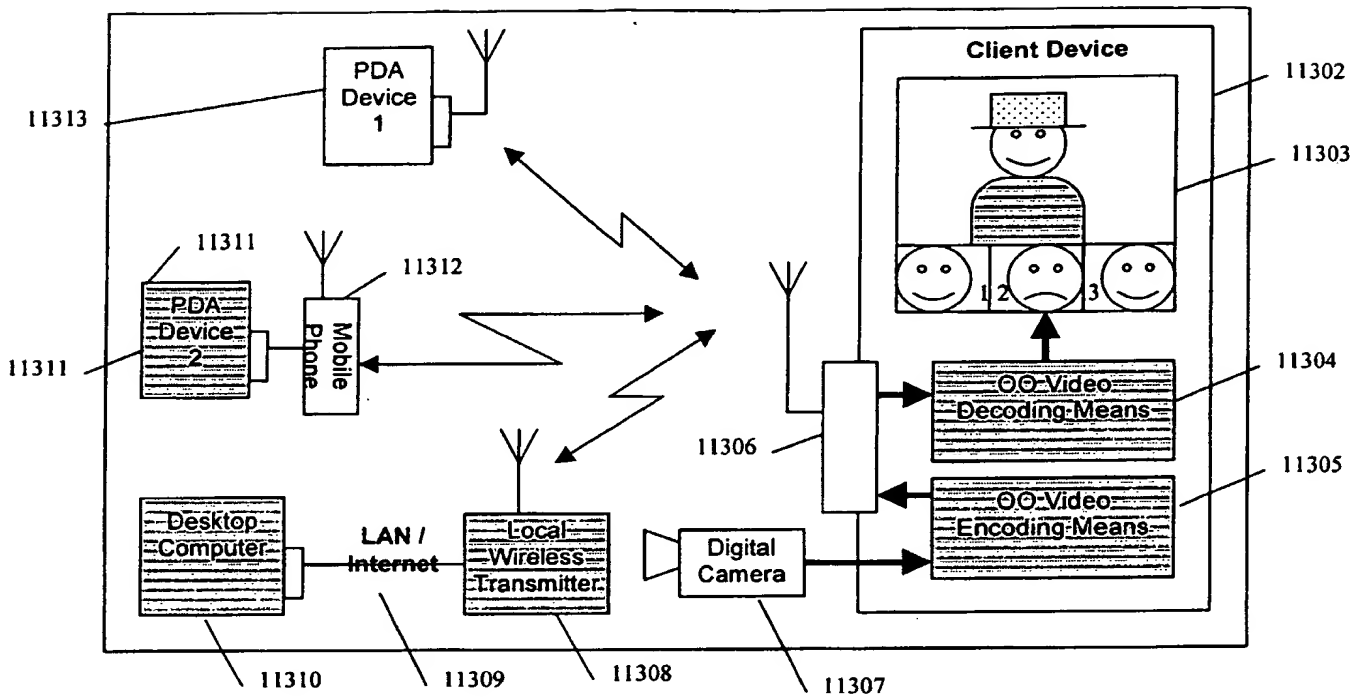
Client Device
11302
11303
11313
PDA Device 1
11311
PDA Device 2
11312
Mobile Phone
11311
11306
OO-Video Decoding Means
11304
OO-Video Encoding Means
11305
Desktop Computer
LAN / Internet
Local Wireless Transmitter
Digital Camera
11310
11309
11308
11307

Figure 35

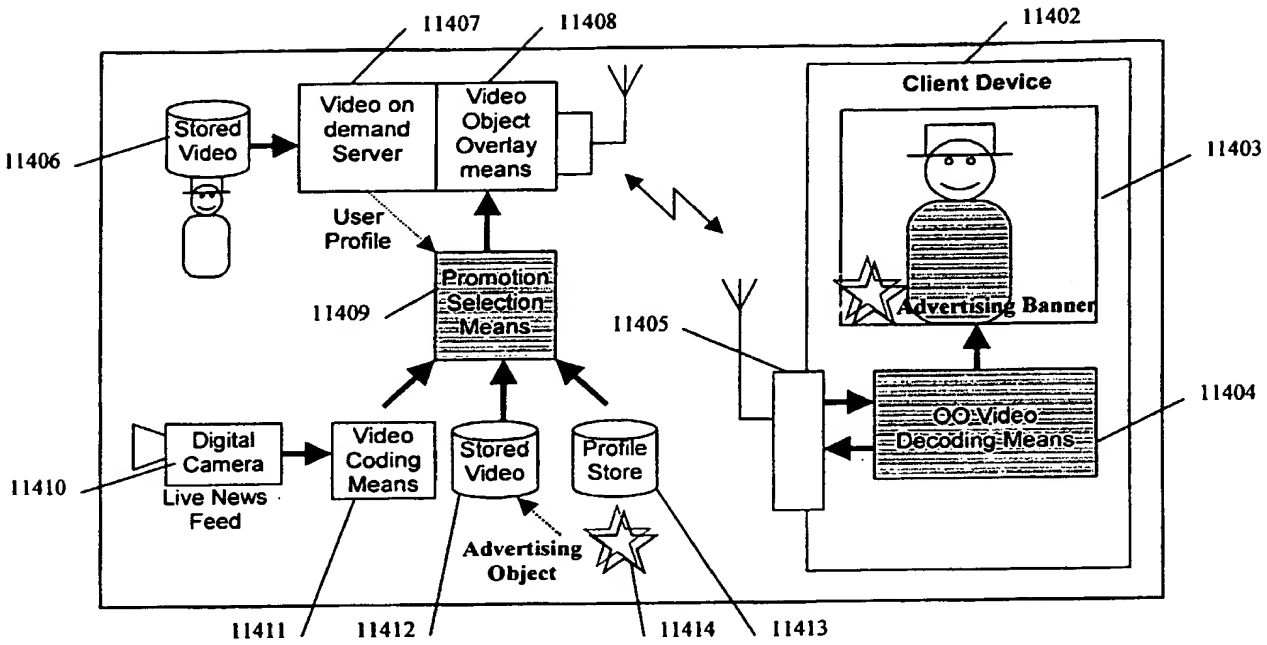
11407
11408
11402
Client Device
Video on demand Server
Video Object Overlay means
Stored Video
11406
11403
User Profile
Promotion Selection Means
11409
Advertising Banner
11405
11404
OO Video Decoding Means
Digital Camera
Live News Feed
11410
Video Coding Means
Stored Video
Profile Store
Advertising Object
11411
11412
11414
11413

Figure 36

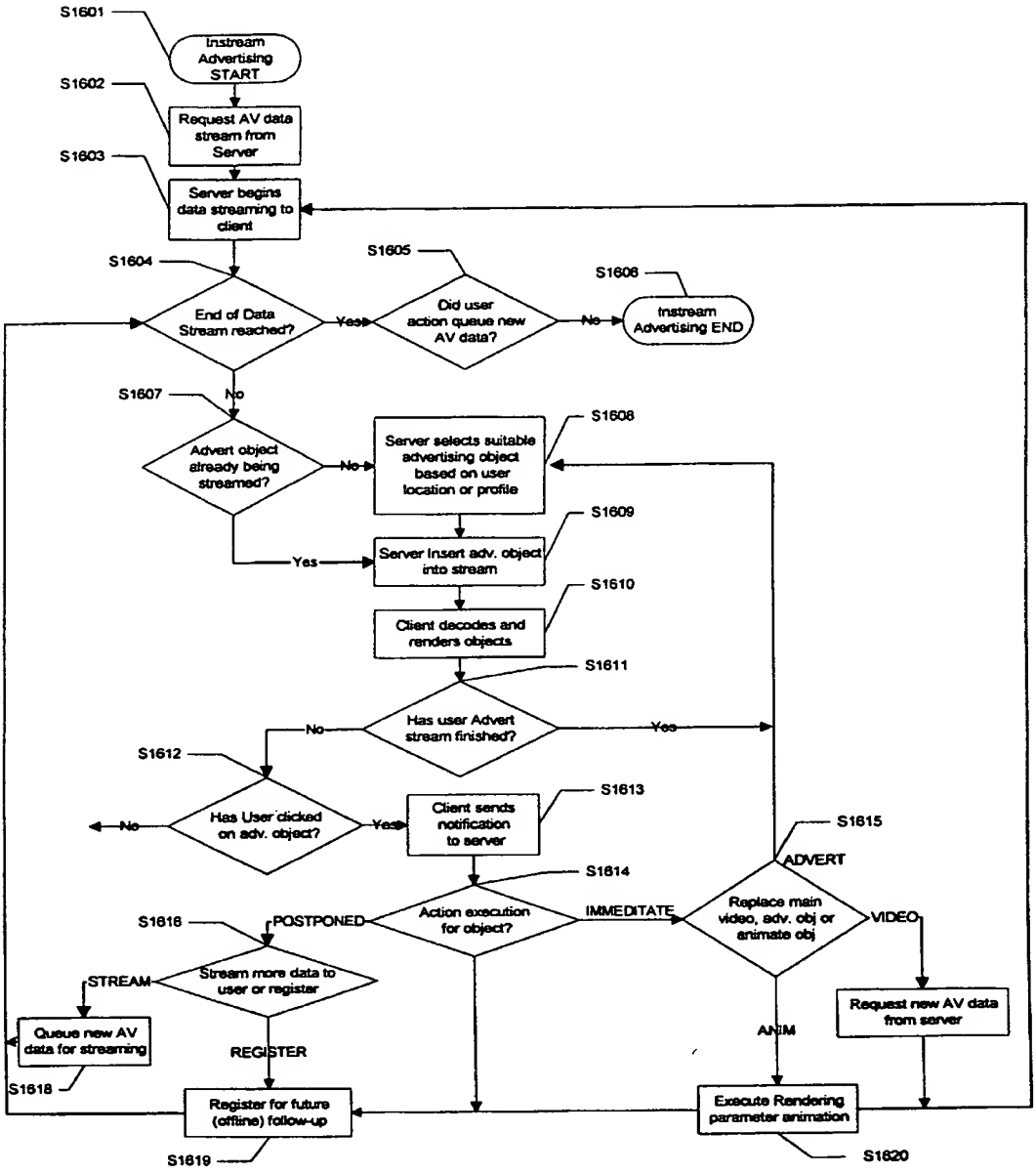
S1601
Instream Advertising START
S1602
Request AV data stream from Server
S1603
Server begins data streaming to client
S1604
End of Data Stream reached?
Yes
S1605
Did user action queue new AV data?
No
S1606
Instream Advertising END
S1607
No
Advert object already being streamed?
No
S1608
Server selects suitable advertising object based on user location or profile
Yes
S1609
Server Insert adv. object into stream
S1610
Client decodes and renders objects
S1611
Has user Advert stream finished?
No
Yes
S1612
Has User clicked on adv. object?
No
Yes
S1613
Client sends notification to server
S1614
Action execution for object?
POSTPONED
IMMEDITATE
S1615
Replace main video, adv. obj or animate obj
ADVERT
VIDEO
ANIM
Request new AV data from server
S1616
Stream more data to user or register
STREAM
REGISTER
Queue new AV data for streaming
S1618
Register for future (offline) follow-up
S1619
Execute Rendering parameter animation
S1620

Figure 37

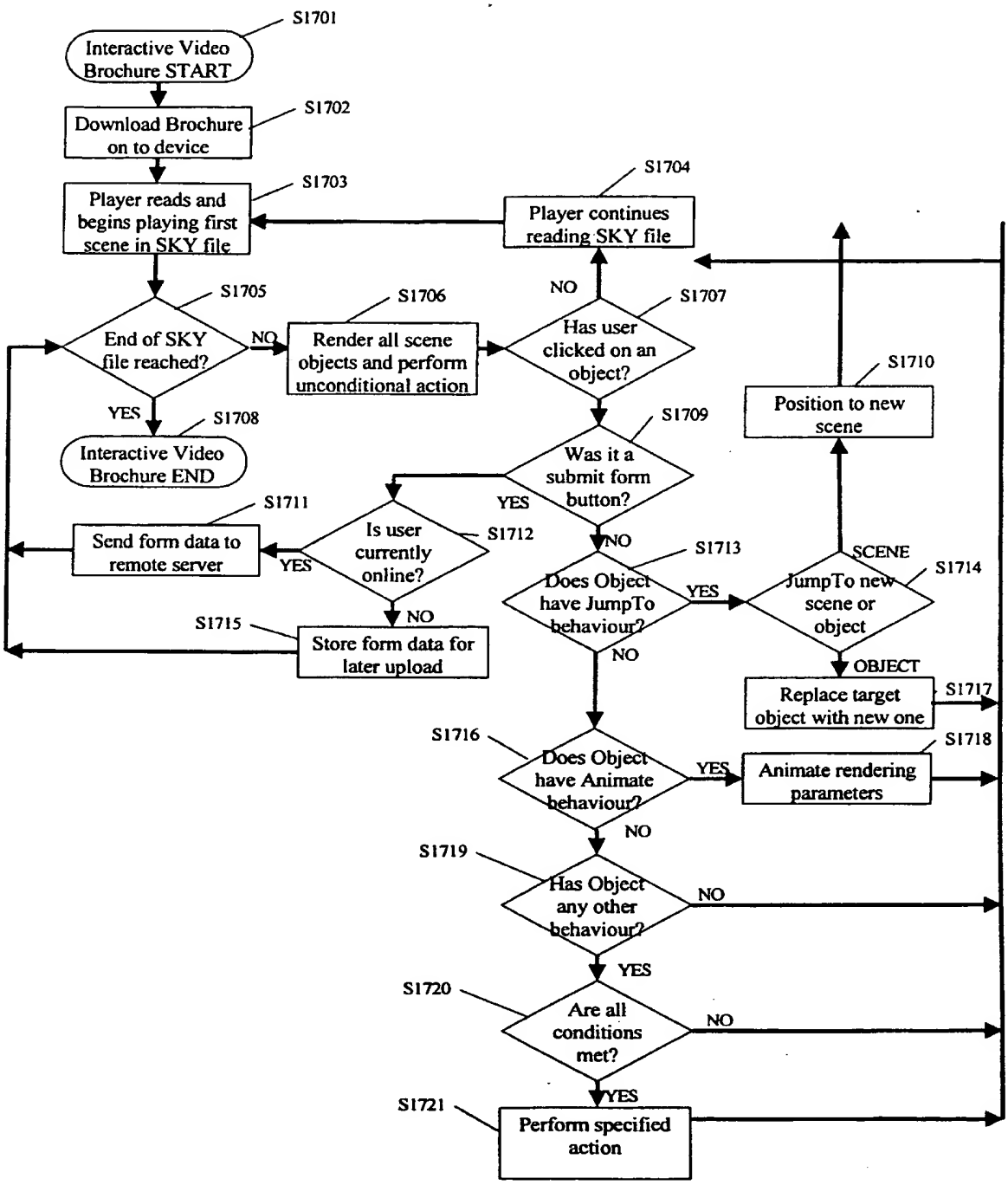
S1701
Interactive Video Brochure START
S1702
Download Brochure on to device
S1703
Player reads and begins playing first scene in SKY file
S1704
Player continues reading SKY file
S1705
End of SKY file reached?
NO
S1706
Render all scene objects and perform unconditional action
S1707
Has user clicked on an object?
NO
YES
S1708
Interactive Video Brochure END
S1709
Was it a submit form button?
YES
NO
S1710
Position to new scene
S1711
Send form data to remote server
S1712
Is user currently online?
YES
NO
S1713
Does Object have JumpTo behaviour?
YES
NO
SCENE
S1714
JumpTo new scene or object
OBJECT
S1715
Store form data for later upload
S1716
Does Object have Animate behaviour?
YES
NO
Replace target object with new one
S1717
S1718
Animate rendering parameters
S1719
Has Object any other behaviour?
NO
YES
S1720
Are all conditions met?
NO
YES
S1721
Perform specified action

Figure 38

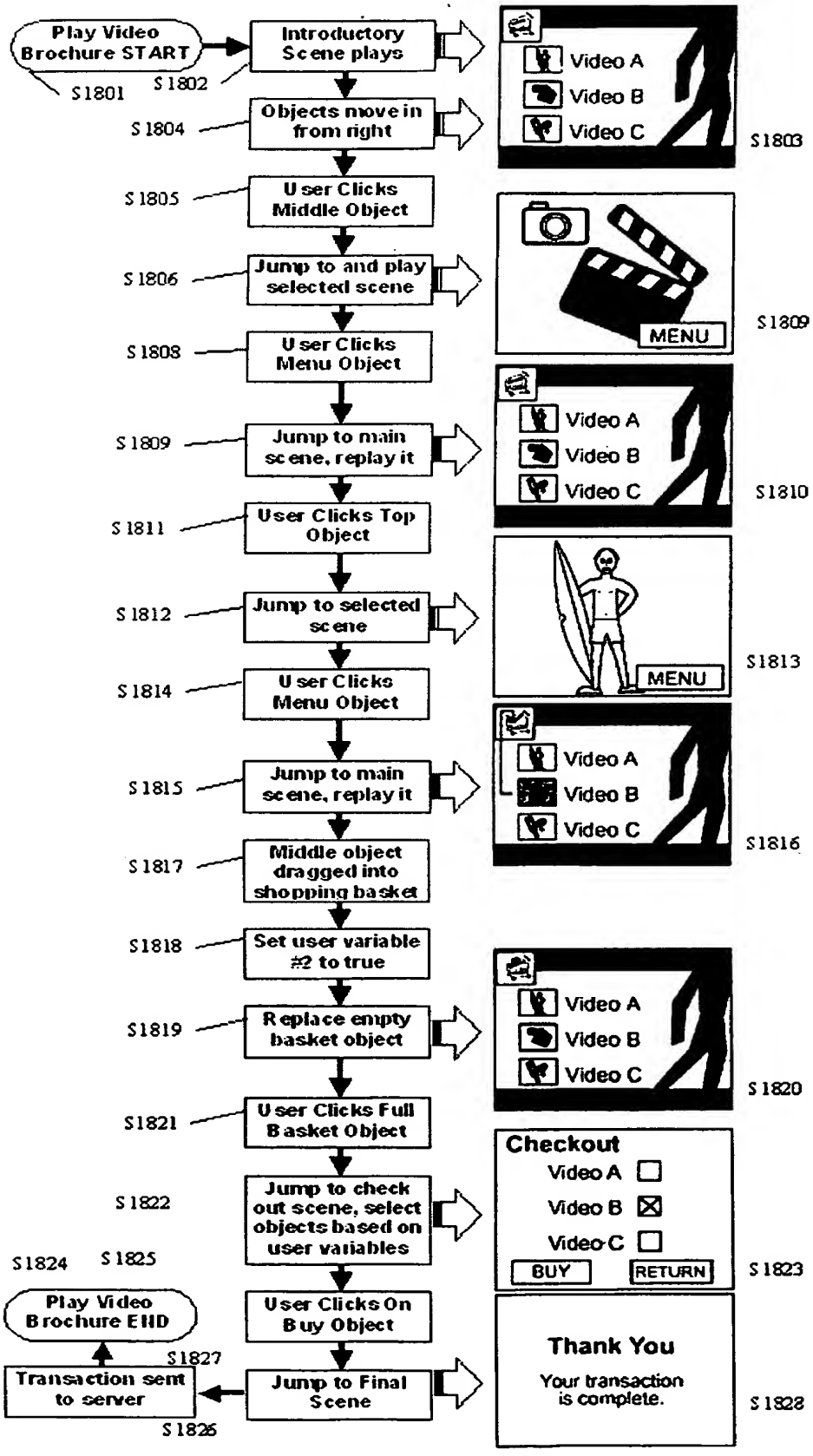
Play Video Brochure START
S1801
Introductory Scene plays
S1802
Video A
Video B
Video C
S1803
S1804
Objects move in from right
S1805
User Clicks Middle Object
S1806
Jump to and play selected scene
MENU
S1809
S1808
User Clicks Menu Object
S1809
Jump to main scene, replay it
Video A
Video B
Video C
S1810
S1811
User Clicks Top Object
S1812
Jump to selected scene
MENU
S1813
S1814
User Clicks Menu Object
S1815
Jump to main scene, replay it
Video A
Video B
Video C
S1816
S1817
Middle object dragged into shopping basket
S1818
Set user variable #2 to true
S1819
Replace empty basket object
Video A
Video B
Video C
S1820
S1821
User Clicks Full Basket Object
S1822
Jump to check out scene, select objects based on user variables
Checkout
Video A
Video B
Video C
BUY
RETURN
S1823
S1824
S1825
Play Video Brochure END
User Clicks On Buy Object
S1827
Transaction sent to server
Jump to Final Scene
S1826
Thank You
Your transaction is complete.
S1828

Figure 39

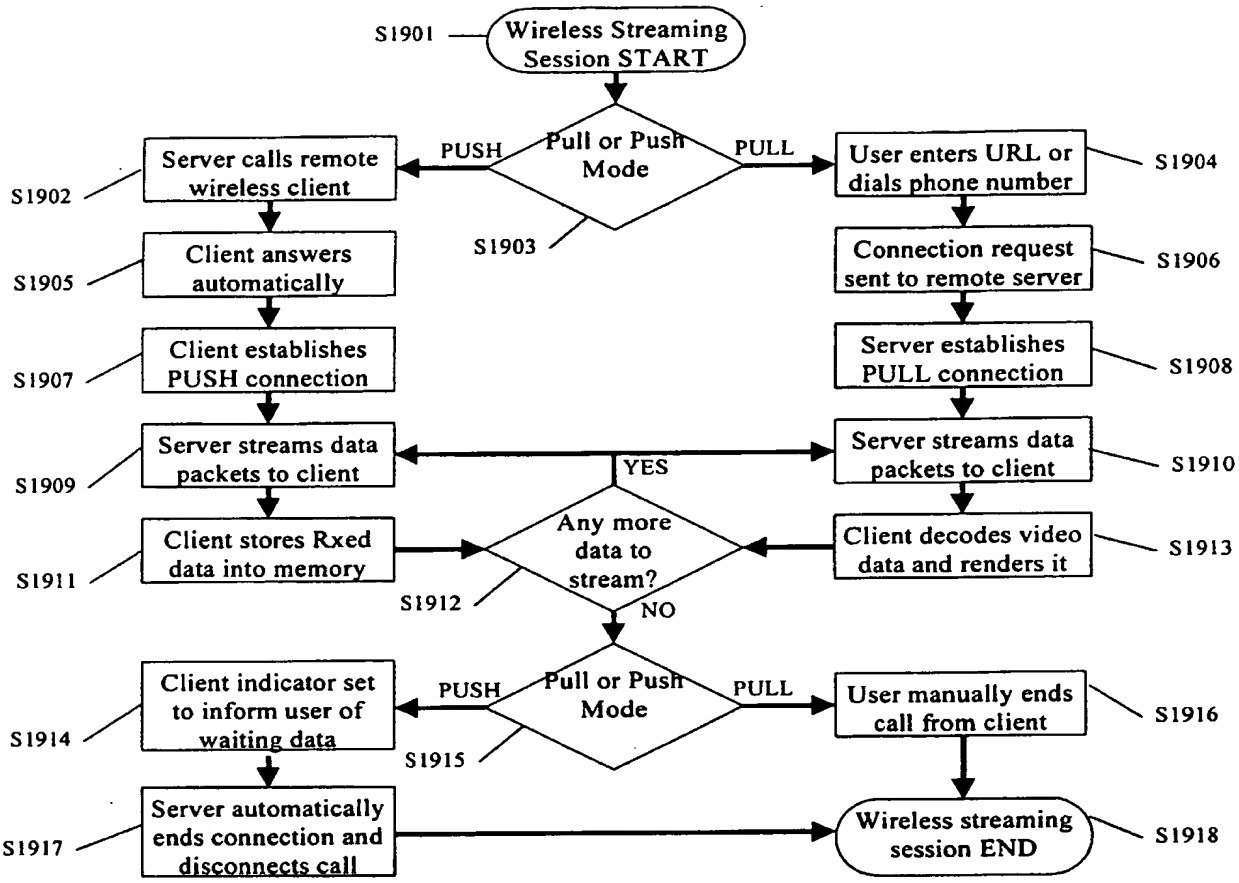
S1901
Wireless Streaming Session START
Pull or Push Mode
PUSH
PULL
S1903
S1902
Server calls remote wireless client
S1905
Client answers automatically
S1907
Client establishes PUSH connection
S1909
Server streams data packets to client
S1911
Client stores Rxed data into memory
S1904
User enters URL or dials phone number
S1906
Connection request sent to remote server
S1908
Server establishes PULL connection
S1910
Server streams data packets to client
S1913
Client decodes video data and renders it
YES
Any more data to stream?
S1912
NO
Pull or Push Mode
PUSH
PULL
S1915
S1914
Client indicator set to inform user of waiting data
S1916
User manually ends call from client
S1917
Server automatically ends connection and disconnects call
S1918
Wireless streaming session END

Figure 40

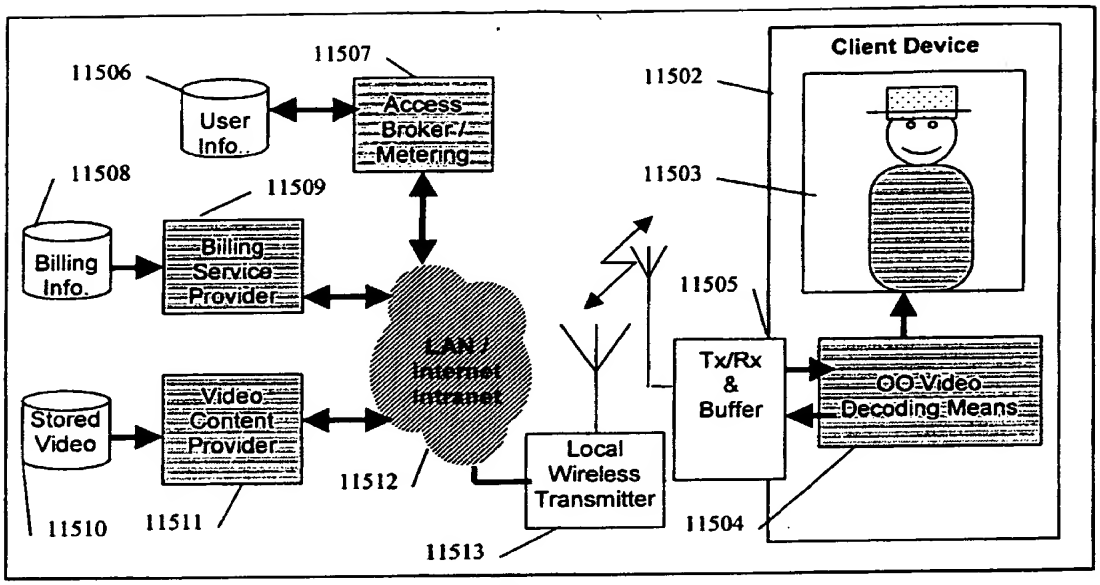
11506
User Info.
11507
Access Broker / Metering
11508
Billing Info.
11509
Billing Service Provider
Stored Video
Video Content Provider
11510
11511
LAN / Internet Intranet
11512
11513
Local Wireless Transmitter
Client Device
11502
11503
11505
Tx/Rx & Buffer
OO-Video Decoding Means
11504

Figure 41

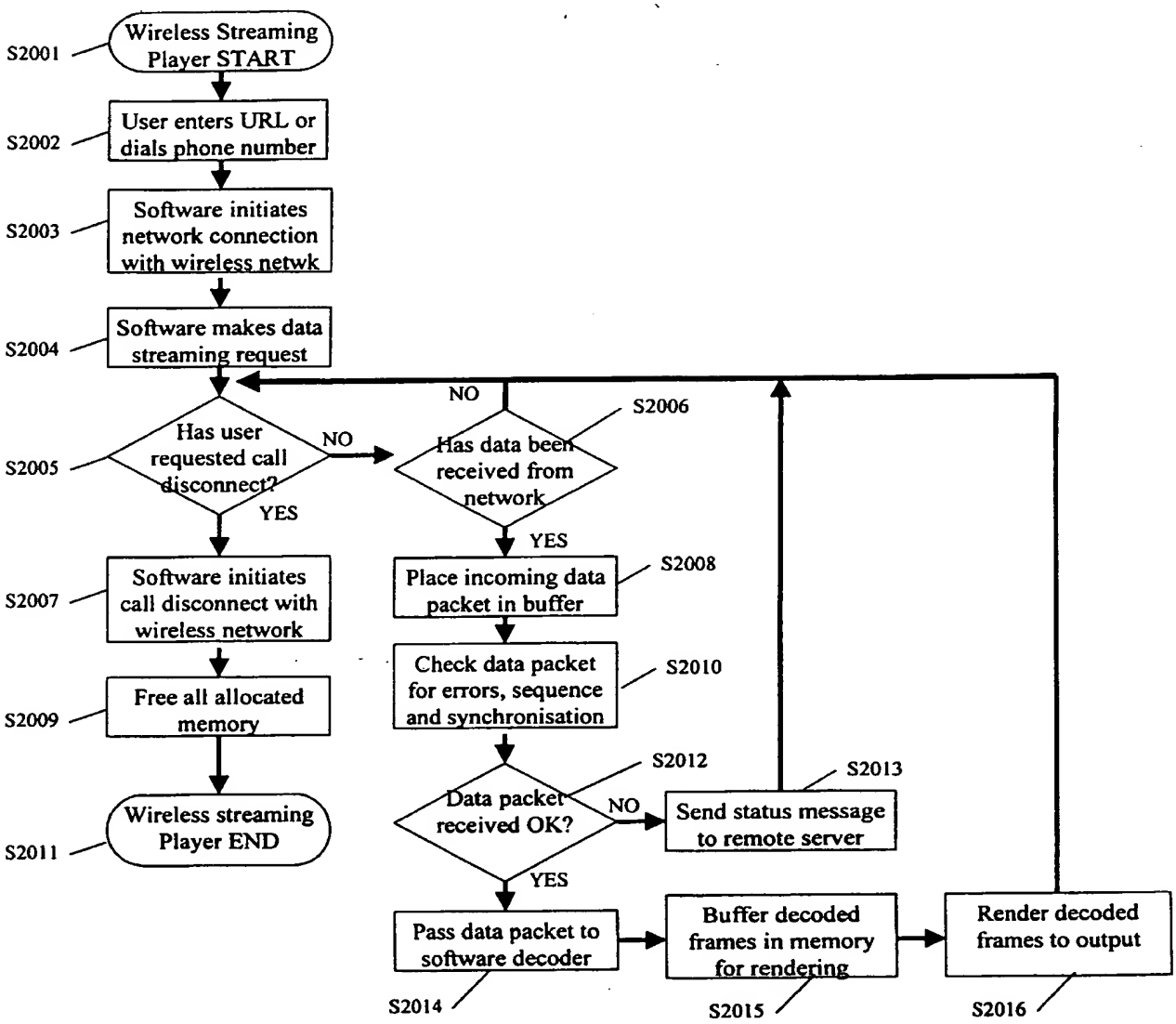
S2001
Wireless Streaming Player START
S2002
User enters URL or dials phone number
S2003
Software initiates network connection with wireless netwk
S2004
Software makes data streaming request
S2005
Has user requested call disconnect?
NO
YES
S2006
Has data been received from network
NO
YES
S2007
Software initiates call disconnect with wireless network
S2008
Place incoming data packet in buffer
S2009
Free all allocated memory
S2010
Check data packet for errors, sequence and synchronisation
S2011
Wireless streaming Player END
S2012
Data packet received OK?
NO
YES
S2013
Send status message to remote server
S2014
Pass data packet to software decoder
S2015
Buffer decoded frames in memory for rendering
S2016
Render decoded frames to output

Figure 42

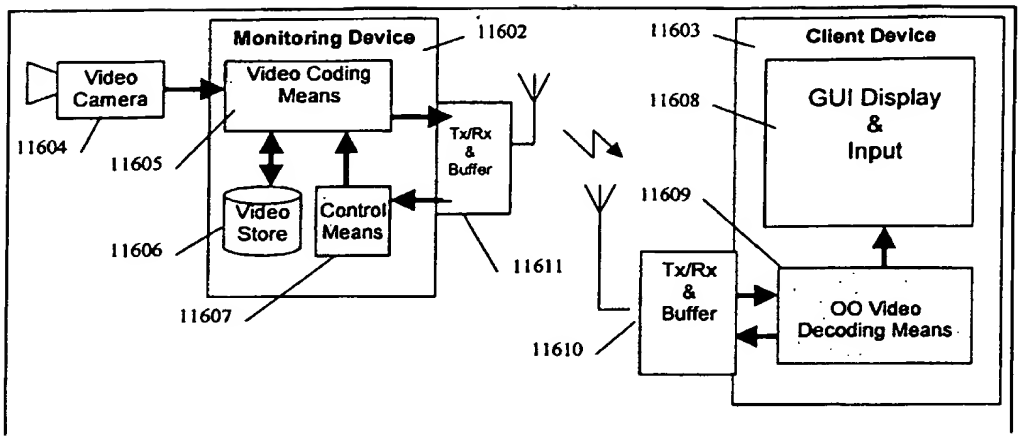
Monitoring Device
11602
Video Camera
Video Coding Means
11604
11605
Tx/Rx & Buffer
Video Store
Control Means
11606
11607
11611
11603
Client Device
11608
GUI Display & Input
11609
Tx/Rx & Buffer
OO Video Decoding Means
11610

Figure 43

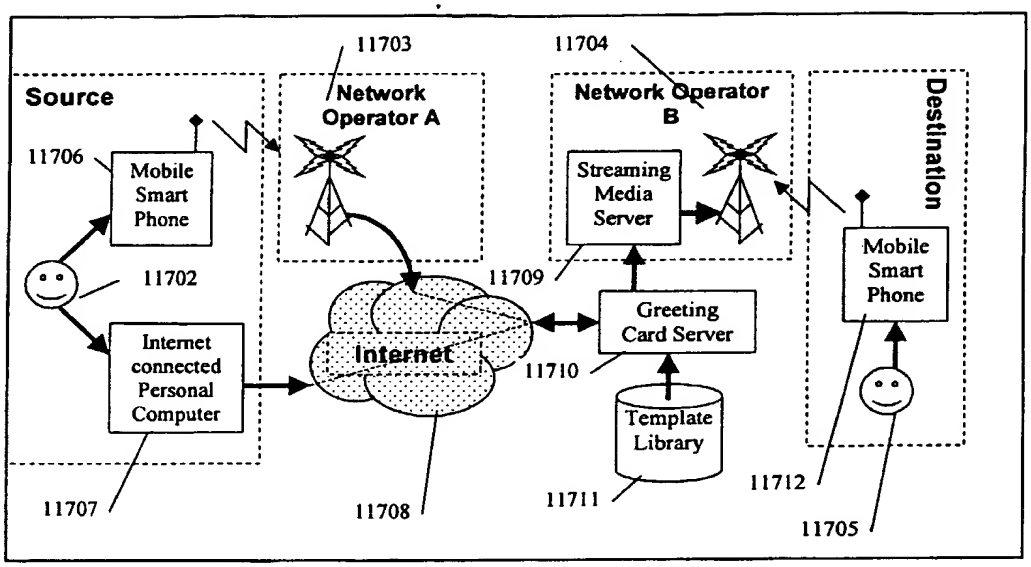
Source
11703
Network Operator A
11704
Network Operator B
Destination
11706
Mobile Smart Phone
Streaming Media Server
Mobile Smart Phone
11702
11709
Greeting Card Server
Internet
Internet connected Personal Computer
11710
Template Library
11707
11708
11711
11712
11705

Figure 44

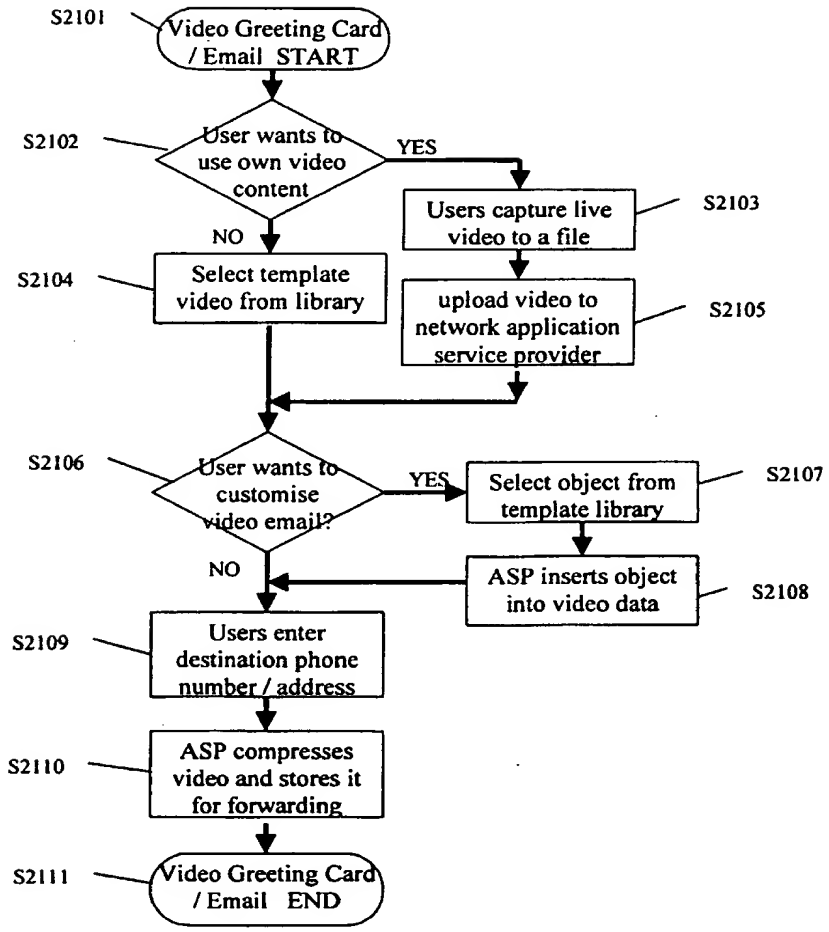
S2101
Video Greeting Card / Email START
S2102
User wants to use own video content
YES
NO
Users capture live video to a file
S2103
S2104
Select template video from library
upload video to network application service provider
S2105
S2106
User wants to customise video email?
YES
Select object from template library
S2107
NO
ASP inserts object into video data
S2108
S2109
Users enter destination phone number / address
S2110
ASP compresses video and stores it for forwarding
S2111
Video Greeting Card / Email END

Figure 45

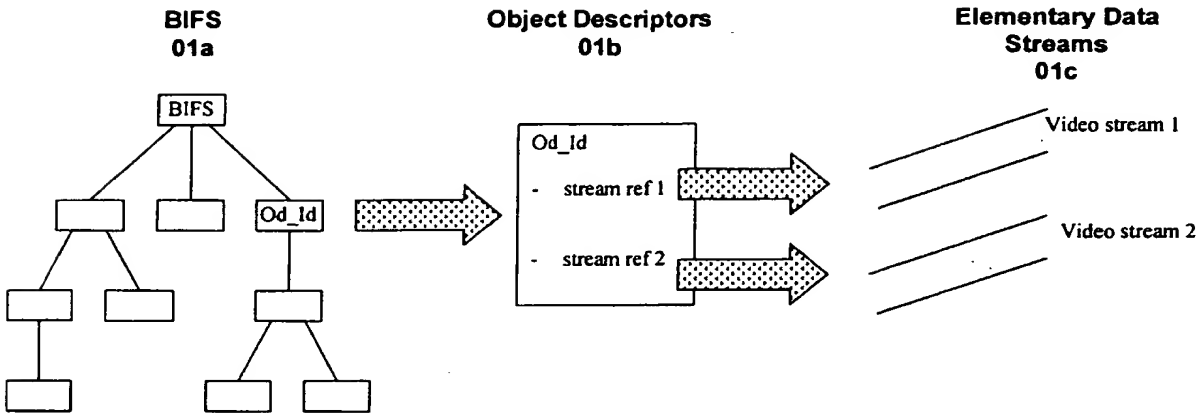

Figure 46

SUBSTITUTE SHEET (Rule 26) - RO/AU

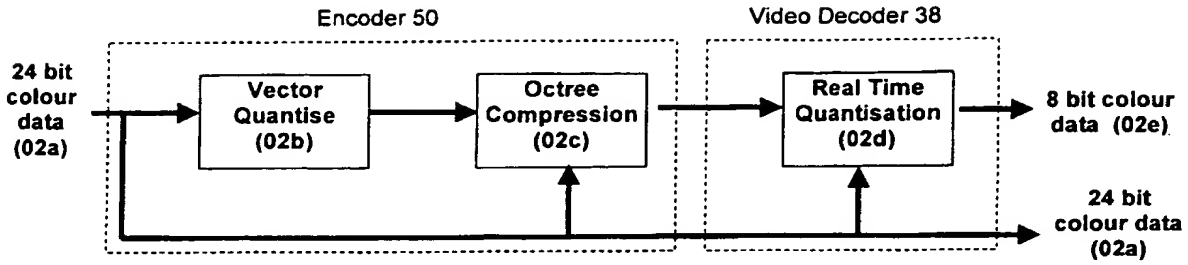
Encoder 50
Video Decoder 38
24 bit colour data (02a)
Vector Quantise (02b)
Octree Compression (02c)
Real Time Quantisation (02d)
8 bit colour data (02e)
24 bit colour data (02a)

Figure 47

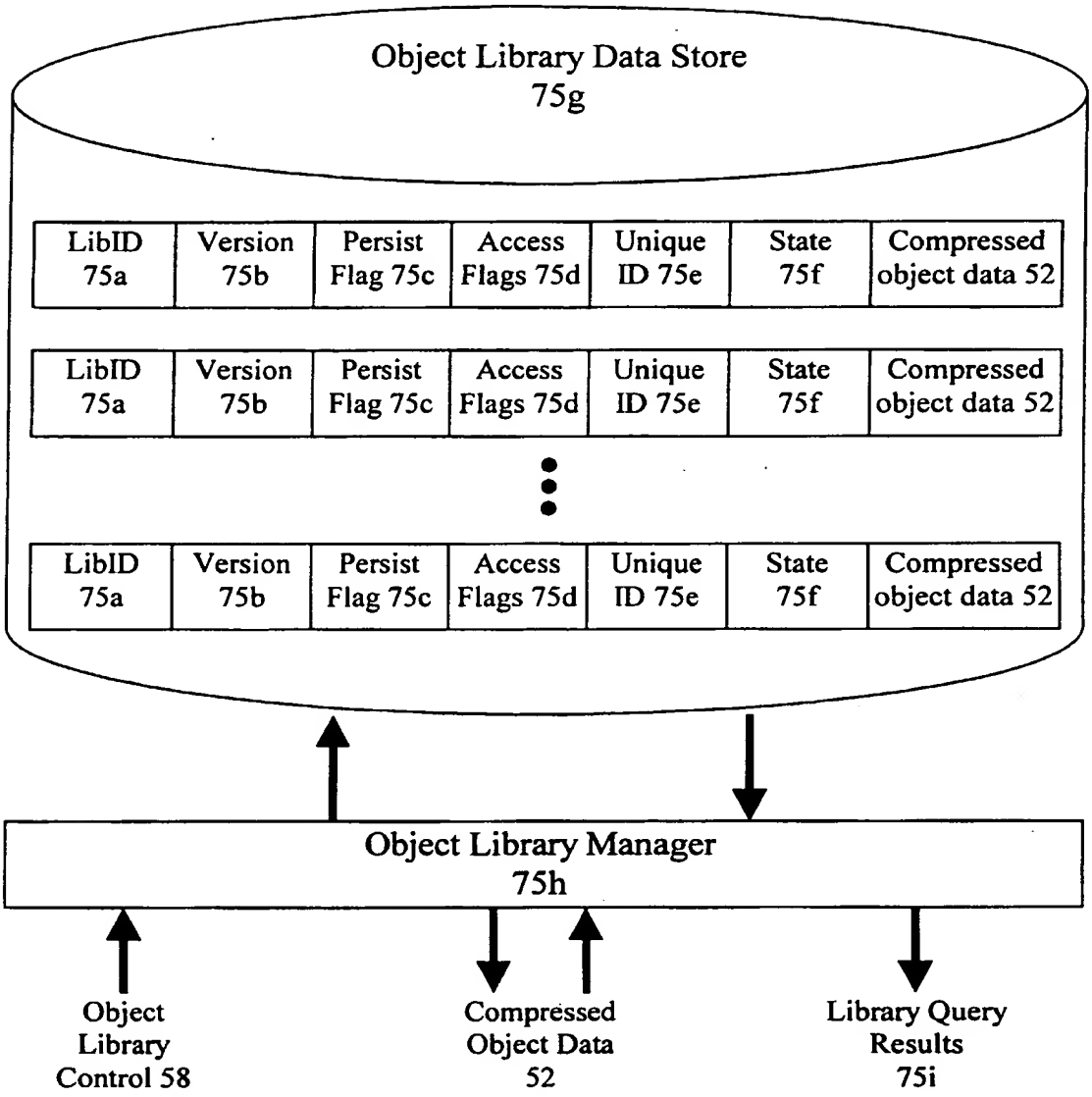
Object Library Data Store
75g
LibID 75a
Version 75b
Persist Flag 75c
Access Flags 75d
Unique ID 75e
State 75f
Compressed object data 52
LibID 75a
Version 75b
Persist Flag 75c
Access Flags 75d
Unique ID 75e
State 75f
Compressed object data 52
LibID 75a
Version 75b
Persist Flag 75c
Access Flags 75d
Unique ID 75e
State 75f
Compressed object data 52
Object Library Manager
75h
Object Library Control 58
Compressed Object Data 52
Library Query Results 75i

Figure 48

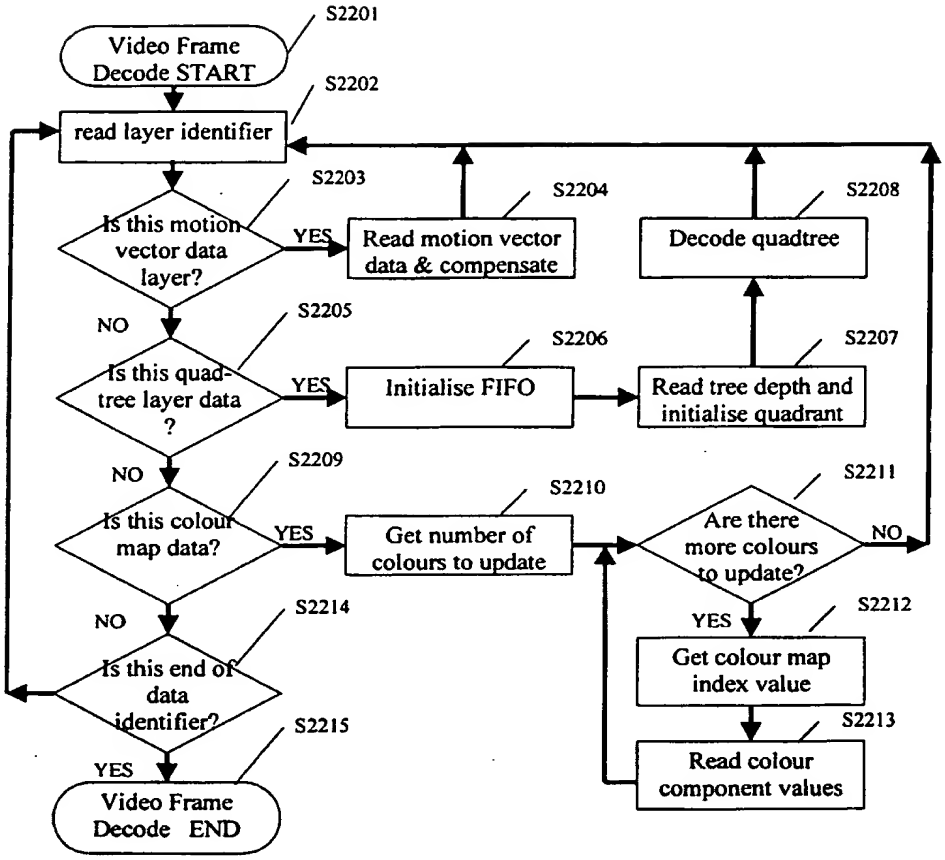
S2201
Video Frame Decode START
S2202
read layer identifier
S2203
Is this motion vector data layer?
YES
S2204
Read motion vector data & compensate
S2208
Decode quadtree
NO
S2205
Is this quad-tree layer data ?
YES
S2206
Initialise FIFO
S2207
Read tree depth and initialise quadrant
NO
S2209
Is this colour map data?
YES
S2210
Get number of colours to update
S2211
Are there more colours to update?
NO
YES
S2212
Get colour map index value
S2213
Read colour component values
NO
S2214
Is this end of data identifier?
YES
S2215
Video Frame Decode END

Figure 49

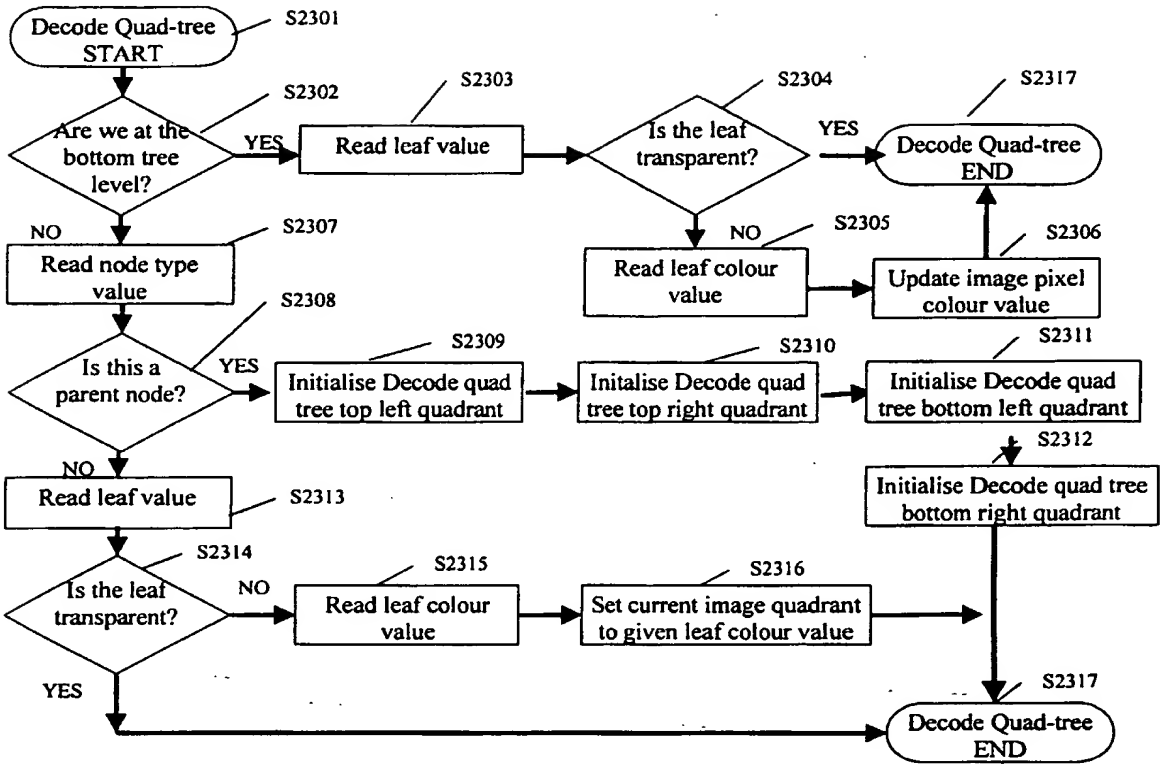

Figure 50

SCANNED, # 12

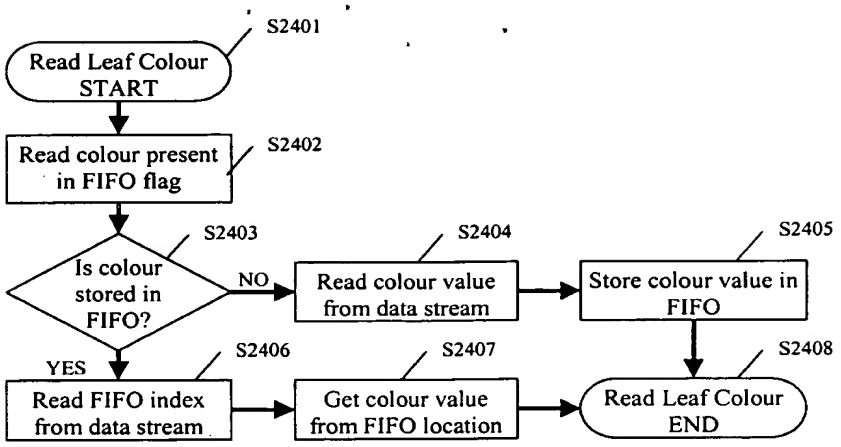
S2401
Read Leaf Colour START
Read colour present in FIFO flag
S2402
S2403
Is colour stored in FIFO?
NO
S2404
Read colour value from data stream
S2405
Store colour value in FIFO
YES
S2406
Read FIFO index from data stream
S2407
Get colour value from FIFO location
S2408
Read Leaf Colour END

Figure 51